University of Alberta

Topology Design for Reconfigurable ATM Networks

by

Chenjiang Hu

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

Department of Computing Science

Edmonton, Alberta

Fall 1997

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-22605-0

Canada

University of Alberta

Library Release Form

Name of Author: Chenjiang Hu

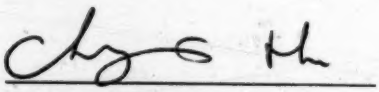
Title of Thesis: Topology Design for Reconfigurable ATM Networks

Degree: Master of Science

Year this Degree Granted: 1997

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material from whatever without the author's prior written permission.



Chenjiang Hu
#102, 10533-83 Avenue
Edmonton, Alberta
Canada T6E 2E1

Date: Sept. 5, 1997

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Topology Design for Reconfigurable ATM Networks** submitted by **Chenjiang Hu** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dr. Janelle J. Harms (Supervisor)

Dr. Pawel Gburzynski (Chair)

Dr. Ioanis Nikolaidis

Dr. Xiaoling Sun (External Examiner)

Date: Sept 5/97

To my wife, Hongying Fan

and my daughter, Lucy Hu

Abstract

ATM is recommended as the preferred transfer mode to support multimedia services in B-ISDN. The logical VP based network provides an effective way to facilitate the dynamic resource management in ATM networks. The physical topology of an ATM network should be designed to provide flexibility to the logical VP reconfiguration in order to adapt to changing traffic conditions and possible network facility failures. In this thesis, we first address the problem of physical topology design for ATM networks with a view to facilitating the logical VP reconfiguration phase. The physical topology design model is formulated as a nonlinear mixed integer programming problem where the objective function to be minimized is defined by considering the balance between the link capacity cost and the link installation cost. Two parameters are introduced in the design model to control the capacity assignments for the primary and alternate routes respectively. Two graph-based heuristic algorithms are developed to solve the problem by exploiting two important properties of the physical topology design model.

In the second part, the performance of generated topologies is studied. The two algorithms are numerically compared with respect to the computational efficiency and the quality of solution. Extensive computational experiments are conducted to investigate topological (static) properties of the generated physical networks. The static performance evaluation indicates that the parameter that controls spare capacity for alternate paths has more effect on the generated topology than the other parameter. The two parameters can be adjusted to achieve better performance with little increase in network cost. Finally, a call-level simulation is carried out to dynamically evaluate the extent to which the generated physical topology provides flexibility to the dynamic bandwidth allocation at the logical VP level. The dynamic performance evaluation shows that the generated physical topology can achieve the same performance with small extra capacity.

Acknowledgements

I would like to thank, first of all, my supervisor, Dr. Janelle Harms, for her consistent encouragement, invaluable guidance and numerous suggestions throughout my study and research in Computing Science. This thesis could not have been completed if it was not for her supervision. It was a great pleasure working with and learning from her.

It is with great pleasure that I would like to thank Dr. Ioanis Nikolaidis and Dr. Xiaoling Sun for their critical reading and valuable comments on my thesis, and for serving as my examining committee members. Thanks also go to Dr. Pawel Gburzynski who chaired my defense.

I wish to express my thanks to all the members in Communication Networks Group. They are all very friendly people to work with. My thanks also go to the department secretaries for their kindness and helpfulness.

Last but not the least, I wish to express my gratitude to Dr. Janelle Harms and the Department of Computing Science at University of Alberta for their financial support.

Contents

1 Introduction 1

1.1 Motivation: Role of VP Subnetworks 1

1.2 Thesis Objective 4

1.3 Outline of Thesis 5

2 Related Work on Topology Design of Communication Networks 6

2.1 A Classification of Topology Design Techniques 6

2.2 Topology Design for Physical Networks 8

2.2.1 Lagrangean Relaxation Method 8

2.2.2 Topology Design with Facility Selection 8

2.2.3 A Branch and Bound Approach 10

2.2.4 A Hierarchical Approach 11

2.2.5 A Genetic Design Algorithm for Local Area Networks 12

2.2.6 A Simulated Annealing-Based Algorithm 14

2.3 Topology Design for Logical Networks 15

2.3.1 Methods for VP Capacity Reallocation 15

2.3.2 Methods for VP Topology Reconfiguration 17

2.4 Topology Design Methods for Survivable Networks 23

2.5 Summary 25

3 A Design Model for Physical Topology in ATM Networks **26**

3.1 Introduction 26

3.2 Problem Formulation 27

3.3 A Graph-Based Heuristic Algorithm 33

3.4 An Alternative Heuristic Algorithm 37

3.5 Two-Connectivity for Link Deletion 39

3.6 Rerouting for Link Deletion 43

4 Experiments for Performance Evaluation **47**

4.1 Comparison for Two Algorithms 47

4.2 Static Evaluation for the Topology Design Model 54

4.2.1 Static Evaluation under Uniform Traffic 56

4.2.2 Static Evaluation under Non-Uniform Traffic 62

4.3 Dynamic Evaluation for the Topology Design Model 66

4.3.1 Dynamic Evaluation under Uniform Traffic 68

4.3.2 Dynamic Evaluation under Non-Uniform Traffic 71

5 Conclusion and Future Work **76**

5.1 Summary and Conclusion 76

5.2 Future Work 79

Bibliography **81**

List of Tables

3.1 Notation 30

4.1 The distance data for EN1 49

4.2 Network Costs and Link Percentages of Selected Topologies Generated by the Two Algorithms for Other Sample Networks 53

List of Figures

3.1 P passes through link e and Q touches C only in vertex w 41

3.2 P passes through link e and Q has several intersections with cycle C 42

4.1 The Comparison of Network Cost between the Two Algorithms . . . 49

4.2 The Comparison of CPU Time between the Two Algorithms 50

4.3 The Comparison of Link Percentage between the Two Algorithms . . 50

4.4 The Layout of Physical Topology for EN-1 under Uniform Traffic with $\rho = 0$ and $\alpha = 0$: (a) Generated by Algorithm 1, (b) Generated by Algorithm 2 51

4.5 The Layout of Physical Topology for EN-1 under Uniform Traffic with $\rho = 0.3$ and $\alpha = 0.1$: (a) Generated by Algorithm 1, (b) Generated by Algorithm 2 52

4.6 The Layout of Physical Topology for EN-1 under Uniform Traffic with $\rho = 0.5$ and $\alpha = 0.5$: Both Algorithms Produce the Same Topology . 54

4.7 Average Link Utilization versus α under Uniform Traffic 56

4.8 Average Link Utilization versus ρ under Uniform Traffic 57

4.9 Maximum Link Utilization versus α under Uniform Traffic 58

4.10 Maximum Link Utilization versus ρ under Uniform Traffic 58

4.11 Average Hop Count of Primary Routes versus α under Uniform Traffic 59

4.12 Average Hop Count of Primary Routes versus ρ under Uniform Traffic 60

4.13 Average Hop Count of Secondary Routes versus α under Uniform Traffic 61

4.14 Average Hop Count of Secondary Routes versus ρ under Uniform Traffic 61
4.15 Link Percentage versus α under Uniform Traffic 63
4.16 Link Percentage versus ρ under Uniform Traffic 63
4.17 Average Link Utilization for NSFNET under Non-Uniform Traffic . . 64
4.18 Average Hop of Primary Routes for NSFNET under Non-Uniform Traffic 64
4.19 Average Hop of Secondary Routes for NSFNET under Non-Uniform Traffic 65
4.20 Link Percentage for NSFNET under Non-Uniform Traffic 65
4.21 Average Call Blocking Rate versus ρ for NSFNET under Uniform Traffic 69
4.22 Average Call Blocking Rate versus α for NSFNET under Uniform Traffic 70
4.23 Average Call Blocking Rate versus α for NSFNET under Non-Uniform Traffic 73
4.24 Average VP Utilization versus α for NSFNET under Non-Uniform Traffic 73
4.25 Average Call Blocking Rate versus ρ for NSFNET under Non-Uniform Traffic 74
4.26 Average Call Blocking Rate versus the Traffic Load under Non-Uniform Traffic 75

Chapter 1

Introduction

1.1 Motivation: Role of VP Subnetworks

The current telecommunication networks are service-oriented. That is, for every individual telecommunication service at least one network exists to transport this service. For example, the public switched telephone network offers the customers classical two-way voice conversation. Computer data are mainly transported in the private domain by LANs in which the famous protocols are Ethernet, token bus and token ring. These specialized networks suffer from three main disadvantages [6]. Each network is only capable of transporting one specific service for which it was designed intentionally, but it is difficultly to adapt it to other services. In this information age, customers are requesting an ever-increasing number of new services and new technology will change the service requirements. A specialized network has great difficulties in adapting to changing or new service requirements. Network resources are used inefficiently since resources available in one network cannot be used by other networks. Taking into account all these disadvantages, it is therefore very important in the future that only a single network exist and that this network be service-independent.

Broadband *integrated services digital network* (ISDN) is expected to support various kinds of services such as: voice, video, data, and image with different traffic characteristics, different qualities of service, and different bandwidth requirements. A transfer mode (telecommunication technique) able to transport such a variety of services must be very flexible in the sense that it must transport a wide range of bit rates which may fluctuate in time. A transfer mode is basically characterized by the switching technique used in the switching nodes of the network. *Asynchronous Transfer Mode* (ATM) is recommended as the preferred transport technology to support such a variety of services on broadband ISDNs [6]. In ATM networks, all kinds of multimedia information are multiplexed and switched using cells with fixed size [39]. They are transported at high speed. The use of a cell-switching technology is a big break with the old tradition of circuit switching within the telephone system. Cell-switching is highly flexible and can handle both constant bit rate traffic (audio, video) and variable bit rate traffic (data) easily. Also at very high speeds in fiber optics, digital switching of cells is easier than using traditional multiplexing techniques. Furthermore, the error protection is omitted due to the high quality of fiber optic links and flow control is not supported so that real time services can be well supported in ATM networks.

ATM networks are connection-oriented. Making a call requires first setting up a *virtual channel connection* (VCC) between the source and destination hosts. The call then uses this connection. Allocating bandwidth and controlling traffic for each virtual channel will require lots of processing time. A powerful transport mechanism for ATM networks is the virtual path concept. By allowing individual VCCs to be handled and switched together as a single group, this transport mechanism results in less processing cost and better VCC admission control. These groups of VCCs are termed as *virtual paths* (VPs). A *VP connection* (VPC) is generally regarded as a directed path in an existing backbone network (i.e., a physical network) between a pair

of nodes, called *VPC terminators*. The nodes on the path between VPC terminators are called *VP transit nodes*. A VCC is just a set of concatenated VPCs. Using the virtual path concept, we can construct a logical network where links correspond to VPCs and nodes correspond to VPC terminators. The logical VP-based network can be designed in such a way that the resources in the ATM network are efficiently used while the *quality of service* (QoS) requirements of all the traffic classes are guaranteed. When the network traffic conditions change, the logical VP-based network can be reconfigured or the VP capacity can be dynamically reallocated to accommodate the current traffic demands.

The extent to which the logical VP reconfiguration is able to improve network efficiency is highly dependent on its ability to provide VCCs with low set-up and switching costs, while maintaining a low call-blocking probability. On the other hand, how to design a logical VP layout or a dynamic capacity reallocation scheme with such an ability is much affected by the topology of the physical network. Let us consider the following three scenarios when a VCC set-up request is made. If a VPC route is available with sufficient capacity to accept the new VCC, the VCC can be established easily with relatively low cost. In fact, the VPCs can provide faster set-up than just using VCs by avoiding processing at transit nodes. (1) If a VPC route exists, but does not have enough capacity to accommodate the new VCC, a dynamic capacity reallocation may be invoked to avoid call blocking. Finally, if no VPC route is found from the source to the destination, a VP reconfiguration is needed to create a new VPC route. The last two scenarios incur higher cost because they result in delays and may require network-wide information to be accomplished. A more serious problem is that the desired change to the VPC topology or capacity allocation may be prevented sometimes due to the restriction of physical topology or link capacity assignments. Therefore, the physical topology of an ATM network should be designed to provide greater flexibility to the logical VP reconfiguration and dynamic capacity reallocation algorithms.

1.2 Thesis Objective

Many algorithms have been reported in the literature for physical topology design in general communication networks. They are often considered as optimization problems with tight link capacity assignments in order to match the expected traffic requirements and to preserve the lowest cost [4] [7] [8] [11] [13] [25]. On the other hand, great interest has been shown in creating a VP reconfiguration and/or developing a dynamic capacity reallocation scheme in order to adapt to changing traffic conditions and to provide fault recovery in the case of link and node failures in ATM networks [9] [12] [27] [19] [29] [32] [33] [36] [42]. The proposed algorithms and schemes for VP reconfiguration and dynamic capacity reallocation mostly assume that the underlying physical network is given. Questions arising naturally are how the underlying physical topology of a ATM network affects the VP reconfiguration algorithms and dynamic capacity reallocation schemes, and how the physical topology for a ATM network can be designed to facilitate the logical VP reconfiguration.

In this thesis, we will consider the problem of physical topology design for ATM backbone networks with a view to facilitating the logical VP reconfiguration phase. To provide flexibility to VP reconfiguration and dynamic reallocation, the physical topology of an ATM network should be robust in the sense that multiple paths exist for each source pair and spare capacity is provided in each link. The objective of our physical topology design is to minimize a weighted function of network set-up cost and the capacity usage cost while maintaining the flexibility for logical reconfiguration. This thesis presents heuristic algorithms to design such a physical topology for ATM networks. Finally, computational experiments are conducted to investigate the effectiveness of our algorithms and how well the generated physical topology facilitates the logical VP reconfiguration phase.

1.3 Outline of Thesis

The remainder of this thesis is organized as follows. In Chapter 2, related work on topology design of general communication networks is reviewed. The topology design can be carried at both physical and logical levels. Most methods proposed for network topology design at both levels are developed based on the formulation of an optimization problem.

Chapter 3 first introduces a design model for physical topology in ATM networks with a view to facilitating the logical VP reconfiguration phase. This physical topology design is formulated as a nonlinear mixed integer programming problem [40]. The objective cost function consists of network set-up cost and the capacity usage cost. Two parameters are included in the optimization problem to make a trade-off between the cost and certain topological properties. Then two graph-based heuristic algorithms are proposed to solve the optimization problem. The complexity of these two algorithms is derived and the theoretical support is given in the last two sections of this chapter.

In Chapter 4, computational experiments are conducted to compare the two algorithms with respect to efficiency and to investigate the effect of the two parameters on the generated physical topology. We also perform a call-level simulation to show the effect of the two parameters in the design model on the dynamic bandwidth allocation at the logical VP level.

Finally, Chapter 5 summarizes the thesis and presents conclusions and some possible directions for future research.

Chapter 2

Related Work on Topology Design of Communication Networks

2.1 A Classification of Topology Design Techniques

According to operational functions, communication networks are typically divided into two categories: local access networks and backbone networks. Most early works dealt with the centralized access network where all communication is to and from a single site. Usually a tree topology is chosen for centralized networks since the other sites have relatively simple equipment not capable of making routing decisions. The backbone networks are built on the top of many local access networks. The topology design problem for a backbone network is more complicated since it typically has a mesh topology. In addition, a logical configuration, based on the physical topology of a backbone network, can be created to improve network operation. In this chapter, we will review the main techniques proposed in recent literature for both physical and logical topology designs for the backbone network.

The physical topology design is to determine the network node sites, select the transmission links interconnecting nodes, and provision the node and link facility. Sometimes it also includes finding routes for all source pairs. Many algorithms for

physical topology design have been developed in the literature since the early 1970s. The problems formulated in the physical topology design are usually stated as an optimization problem with tight link capacity assignments in order to match the expected traffic requirements and give the lowest cost. Various optimization techniques are employed to solve the design problem [4] [7] [8] [11] [13] [25]. A quite different goal for physical topology design is to provide the survivability (fault tolerance) to the network. Most methods for survivable network design are to determine spare capacity and/or alternate routes for the network facility (nodes and links) failures [15] [36] [46]. The survivable network design can also be carried out in the logical level [35] [36] [46].

The logical topology design is considered in ATM networks based on the VP concept. It is to provide efficient resource management (e.g., speeding up connection set-up and simplifying routing at transit node) and dynamic traffic control (e.g., adapting to varying traffic and facility failures) to the network. Depending on the extent to which the traffic fluctuates in the network, the VP topology design activities can be performed in two different ways: capacity reallocation and VP topology reconfiguration. The capacity reallocation can be used to adjust the changes in capacities allocated to specific VPCs under small traffic variation. The algorithms for VP capacity reallocation are developed in a distributed manner with local information and local objective [2] [44] [45]. For large fluctuations in network traffic, the VP topology reconfiguration should be performed, which includes making network-wide changes to the topology and capacity assignments. The VP topology reconfiguration problem is typically formulated to optimize some global control objective and usually results in a complex and time-consuming algorithm to be implemented in a centralized manner [9] [12] [27] [19] [29] [32] [33] [36] [42].

The notation in this chapter is adopted from corresponding papers, which is independent to those in the later chapters.

2.2 Topology Design for Physical Networks

This section presents a review for recent design methods for the physical topology design mainly for backbone networks but also some methods for both backbone and local access networks.

2.2.1 Lagrangean Relaxation Method

By allocating link capacity tightly and providing single route for each communication pair, Gavish [11] considers the topology design problem for computer communication networks. The topology design is to determine where to place network control processors (NCPs), select the set of backbone links to connect NCPs and the set of access links to connect the end user nodes with the NCPs, and find the set of routes which support communication between end user node pairs. The objective of the design is to balance the overall investment in the network versus the queueing costs imposed on the network users. The design problem is formulated as a nonlinear combinatorial optimization problem. An effective solution procedure for a Lagrangean relaxation of the problem is presented and further improved using subgradient optimization techniques. Three heuristics are developed for generating feasible solutions and demonstrated on problems involving 200 end users and up to 30 NCP locations.

2.2.2 Topology Design with Facility Selection

Gersht and Weihmayer [13] consider joint optimization of physical topology design and facility selection. The objective is to minimize total network cost while selecting facility types, allocating capacity, and routing traffic to accommodate demand and performance requirements. They assume that a single facility type is allowed on each link. A graph with N nodes is considered and all links in the graph are potential candidates for the network design. The link cost C^t_{mn} depends on the capacity α^t_{mn}

for link (m,n) and facility type t and is the following concave function with startup and incremental costs:

$$C_{mn}^t = \begin{cases} 0 & \text{if } \alpha_{mn}^t = 0 \\ A_{mn}^t + B_{mn}^t \alpha_{mn}^t & \text{if } \alpha_{mn}^t > 0 \end{cases} \tag{2.2.1}$$

where $A_{mn}^t \geq 0$ is the facility-link startup cost and $B_{mn}^t \geq 0$ is the incremental cost. The nodal cost C_m at node m only depends on the nodal capacity β_m:

$$C_m = D_m \beta_m \tag{2.2.2}$$

where D_m is the capacity cost per unit at node m. The total network cost is written

$$z = \sum_{t,mn} (A_{mn}^t \eta_{mn}^t + B_{mn}^t \alpha_{mn}^t) + \sum_m D_m \beta_m \tag{2.2.3}$$

where η_{mn}^t are the decision variables and take value of 1 if facility t on link (m,n) is activated and 0 otherwise.

It was assumed that α_{mn}^t and β_{mn}^t are continuous variables and are restricted by some upper bounds. The average packet delay is calculated by modeling the networks as a network of independent $M/M/1$ queues. The performance constraint is the average packet delay less than some fixed number and the reliability constraint specifies the minimum number of arcs at a node. Thus the joint optimization is defined to minimize (2.2.3) under the above constraints and the routing flow constraints, which is a mixed integer programming problem.

They show that the original problem can be decomposed into two subproblems solved sequentially in the following two steps:

- minimum cost topological design, facility selection, and routing;
- optimal capacity calculation.

The first subproblem is solved based on a heuristic greedy algorithm. The heuristic greedy algorithm first employs a link reduction method to find a topology and then

allocates traffic on minimum cost routes. For the second subproblem, they prove that the optimal capacity assignment leads to a balanced network with all links of the same facility type having the same loading.

However, the tight capacity assignment and single route are considered in the optimization problem so that the physical network will not provide flexibility to the dynamic resource management. The facility selection adds more complexity to the design problem.

2.2.3 A Branch and Bound Approach

Chattopadhyay, Morgan and Raghuram [4] propose an innovative technique for the backbone network design. The objective is to find a network with minimum cost given: 1) number of host sites; 2) traffic between each pair of sites; 3) permissible communication delay between each pair of sites; and 4) a set of choices of communication channels capacities. Let N be the number of host sites and C_{max} be the maximum number of available capacities from which to choose. Denote x_{ij} to be the capacity variable of link (i,j), the domain of which is $\{0,1,\cdots,C_{max}\}$. Let w_{ij} be the routing variable for a source-destination pair. If $w_{ij}=k$, it means that the first stop of the route between nodes i and j is the node k. They assumed that each pair has only one path as the traffic route.

The cost function consists of two parts. The first part is the cost of establishing communication lines, which is expressed as follows:

$$C_1 = \sum_{i \neq j}^{N}((C_c^f + C_c^v D_{ij})|x_{ij}=c) \tag{2.2.4}$$

where C_c^f and C_c^v are the fixed cost and variable cost of installing a link with capacity Q_c for $c=0,1,\cdots,C_{max}$ and D_{ij} is the distance between nodes i and j. The second part is the communication link utilization cost, expressed as:

$$C_2 = \sum_{i \neq j}^{N} \sum_{k,l} (\psi_{ij} C_{kl}^{com} | kl \in \text{path}(i,j)) \tag{2.2.5}$$

where C_{kl}^{com} is the communication cost of link kl and ψ_{ij} is the traffic rate between i and j. The constraints are similar to the above.

They solved the problem as two steps. A heuristic branch and bound method is used to determine the topology (solve for x_{ij}) in the first step. Routing variables are then selected using Ford-Fulkerson's algorithm. It is noted that the first step is very time consuming when N is moderately large since the branch and bound method is generally exponential function of N.

2.2.4 A Hierarchical Approach

Lee, Ro and Tcha [25] tackle the topology design for a two-level hierarchical network where the upper-level hub network is of ring type and lower-level local access networks are of star-type. The following assumptions were made in the problem. 1) The sites of potential hub nodes and user nodes are known. 2) There is no limit on arc capacities and on the number of user nodes that can be assigned to each established hub node. 3) Each pair of user nodes may be associated with two different commodities by having the roles of origin and destination reversed, and each commodity may have a different flow requirement. 4) All flows are unidirectional. 5) Three different types of cost elements are considered: variable flow cost and two kinds of fixed costs: one for establishing hub nodes, and the other for establishing arcs between a pair of hub nodes, or between a hub node and a user node.

Let N be the set of nodes and A be the set of arcs for a network. The node set N consists of two types of nodes: one for candidate hub nodes, N_1, and the other for user nodes, N_2. The arc set A is partitioned into two sets, A_1 and A_2, so that A_1 represents directed arcs between candidate hub nodes and A_2 represents undirected arcs between user nodes and candidate hub nodes. Let K denote the set of commodities. The network design problem is formulated as the following mixed

0-1 integer programming model:

$$\min Z = \sum_{j \in N_1} f_j z_j + \sum_{(j,l) \in A_1} d^1_{jl} y^1_{jl} + \sum_{(i,j) \in A_2} d^2_{ij} y^2_{ij} + \sum_{k \in K} \{ \sum_{(j,l) \in A_1} c^k_{jl} x^k_{jl} + \sum_{(i,j) \in A_2} (c^k_{ij} x^k_{ij} + c^k_{ji} x^k_{ji}) \}, \tag{2.2.6}$$

under the usual traffic flow constraints, where z_j is the 0-1 variable concerning the establishment of the hub node j, y^1_{jl} (y^2_{ij}) is the 0-1 variable concerning the establishment of directed $arc(j,l)$ for the upper level ring network (undirected $arc(i,j)$ for the lower level network), x^k_{jl} (x^k_{ij} or x^k_{ji}) is the nonnegative variable denoting the amount of flow of commodity k on directed $arc(j,l)$ (undirected $arc(i,j)$), f_j is the fixed cost of establishing hub node j, d^1_{jl} (d^2_{ij} is the fixed cost of establishing directed $arc(j,l)$ (undirected $arc(i,j)$), and c^k_{jl} (c^k_{ij}) is the per unit arc flow cost of commodity k on directed $arc(j,l)$ (undirected $arc(i,j)$).

Based on the special configuration, they first consider the dual problem of (2.2.6) and develop a dual ascent procedure to obtain a solution. Then a heuristic is developed to construct a primal feasible solution from the dual solution, which is essentially made of search and local improvement.

The design model results in tight capacity assignments in the links and provides only one route for each source pair so that it is difficult to reconfigure the network topology to adapt to changing traffic conditions. Also they incorporate two levels of networks in the model, which makes the problem more complicated.

2.2.5 A Genetic Design Algorithm for Local Area Networks

Elbaum and Sidi [7] develop a genetic algorithm for topological design in local area networks. They consider a local area network (LAN) that connects N stations (users). The communication traffic demands among the users are denoted by an $N \times N$ matrix A. It is assumed that the LAN is partitioned into P clusters. The $N \times P$ clustering

matrix R specifies which user belongs to which cluster, i.e.,

$$r_{ij} = \begin{cases} 1 & \text{if } \textit{user } i \text{ is in } \textit{cluster } j \\ 0 & \text{otherwise} \end{cases} \tag{2.2.7}$$

Constraints $\sum_{j=1}^{P} r_{ij} = 1, i = 1, \cdots, N$ specify that a user can only belong to one cluster. The cluster traffic matrix S represents the traffic between any two clusters and is equal to $R^T AR$. They assume that network configurations are limited to spanning tree topologies. Let x_{ij}^k be 1 if traffic from cluster i to j passes through cluster k and be 0 otherwise. Let y_{ij}^{kl} be 1 if traffic from cluster i to j passes existing bridge connecting clusters k and l and be 0 otherwise. Base on the above decision variables and traffic matrices, they define the total offered traffic,

$$\Gamma = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} = \sum_{i=1}^{P}\sum_{j=1}^{P} s_{ij},$$

the total traffic at cluster k,

$$L_k = \sum_{i=1}^{P}\sum_{j=1}^{P} s_{ij} x_{ij}^k, 1 \le k \le P,$$

and the total traffic through bridge (k, l),

$$F_{kl} = \sum_{i=1}^{P}\sum_{j=1}^{P} s_{ij} y_{ij}^{kl}, 1 \le k, l \le P.$$

The M/M/1 model is used to describe a single cluster behavior. The total average delay in the LAN is the objective function and consists of the delays of the segments and the bridges:

$$D = \frac{1}{\Gamma}\{\sum_{k=1}^{P} \frac{L_k}{C_k - L_k} + \sum_{i=1}^{P}\sum_{j=1}^{P} F_{ij} B_{ij}\}, \tag{2.2.8}$$

where C_k is the capacity at cluster k and B_{ij} is the delay per packet at the bridge (i, j).

A heuristic genetic algorithm is used to find the minimum of (2.2.8). A sparse Huffman tree is used as the representation of the spanning tree structure. Genetic algorithms are essentially the search procedures based on the mechanics of natural

selection and natural genetics. The three basic operations in genetic algorithm are reproduction, crossover and mutation. Elbaum and Sidi [7] also obtain a lower bound of (2.2.8) to compare with results of the genetic algorithm solution. This method cannot be used to design the topology for the reconfigurable ATM networks since it does not provide multiple routes and extra capacity.

2.2.6 A Simulated Annealing-Based Algorithm

Ersoy and Panwar [8] propose a simulated annealing-based algorithm for finding a minimum delay spanning tree topology as an interconnected LAN/MAN network. This method searches for a better solution at the neighborhood of the current tree. Given a spanning tree, a neighbor tree is created by removing a branch, resulting in two separate subtrees, and then adding another branch which will connect the subtrees, but will not create a loop. Each new topology is evaluated for the average delay (2.2.8) and the constraints. If the transition from the current tree to a new tree is delay-decreasing one, it is always accepted. If the transition is a delay-increasing one, the new tree is accepted as the current solution with the probability p:

$$p = \exp\left(\frac{-[Delay(\text{New Tree}) - Delay(\text{Current Tree})]}{c}\right),$$

where c is the cooling constant.

Ersoy and Panwar [8] also develop a lower bound for the average network delay. Since the average delay (2.2.8) is a combination of delays due to LAN's and bridges, they first find the lower bound of the two parts. The lower bound is found by obtaining the lower bound on the total flow on all bridges through solving the optimum requirement spanning tree problem and by then finding the optimum distribution of the total flow among all bridges such that the average bridge delay is minimized. The same method applies to the second part: the average LAN delay.

The simulated annealing technique is very effective to find a solution near to the global minimum. However, the design model does not include any requirements to make the network topology flexible for dynamic resource management.

2.3 Topology Design for Logical Networks

The logical topology design is needed to provide an efficient resource control as reacting to changes in forecasting information and/or component failures. In ATM networks, this corresponds to finding an overlay of VPCs. This section presents a review of various methods for VP capacity reallocation and VP topology reconfiguration although it is not comprehensive. A survey of resource management with VPs in ATM networks can be found in [10].

2.3.1 Methods for VP Capacity Reallocation

A Dynamic VP Bandwidth Reallocation Method

Baba, Murata and Miyahara [2] propose a dynamic VP bandwidth reallocation method, which dynamically controls the bandwidth allocated to each VP according to traffic fluctuation. This method is used for the case that multiple VPs are multiplexed on the same link between two nodes. When the spare capacity in a VP is below the prespecified threshold level, the VP requests an additional bandwidth. The amount of incremented bandwidth at each request is a prespecified step-size value. However, when a VP has extra capacity more than the threshold level, assigned bandwidth of the VP is released by a step-size value but never falls below its initial bandwidth. An approximate analysis for the method is provided and the validation of its accuracy is assessed by comparing with simulation results. Through numerical examples, they show that this dynamic method can provide performance improvement in terms of call blocking probability.

A Successive VP Bandwidth Modification Method

Shioda and Uose [44] introduce a successive modification concept for VP bandwidth control in ATM networks, which is to absorb the difference between the estimated traffic and the traffic that is actually offered. This successive modification method is to use multiple distributed controllers that use only local information under short time period. The bandwidth is modified successively as these distributed controllers communicate with each other. Two algorithms are developed for two control objectives: minimizing the worst blocking probability and maximizing the weighted carried traffic, respectively. Numerical evaluation using a sample network has shown the effectiveness of two algorithms for preventing the performance degradation caused by large-scale traffic imbalance in the network. Comparison with the batch modification method, which has no feedback effect, shows that the successive modification method with appropriate control intervals can be more responsive to traffic variation over time, but is slightly less responsive when the traffic does not change with time.

A Sizing and Provisioning Method for ATM Networks

Shioda, Saito and Yokoi [45] consider the sizing problem in designing VP-based networks and transmission-path (TP) networks. The notion of bandwidth demand is introduced to capture changing traffic conditions in ATM networks. The bandwidth demand of a VP is defined as the bandwidth needed to carry the traffic load so that the QoS requirements of VCs multiplexed in this VP are satisfied. Based on periodically measured traffic load, the bandwidth demand is adjusted by removing or adding some bandwidth to keep the QoS at a prespecified level. The bandwidth demand of TP is defined as the sum of demands over all the VPs passing the TP plus the maximum demands used for all the back-up VPs used for rerouting under a TP failure. They consider the TP provisioning problem for a TP group (TPG) which contains all the TPs having the same terminators. The TP provisioning procedure

proposed is based on the TP utilization which is a measure of TP congestion. The residual lifetime of a TPG is defined as the time, when the capacity in the TPG no longer accommodate the TPs in the group, minus the time when the utilization of the TPG exceeds a threshold value. By using the Cox proportional hazard model for the lifetime of a TPG, the bandwidth of the TPG is provided to keep the specified utilization level with changes in TP bandwidth demands.

2.3.2 Methods for VP Topology Reconfiguration

Logical Reconfiguration for Minimizing the Average Number of Packets

Lee and Yee [27] [29] consider the logical topology design for reconfigurable networks. A reconfigurable network is a circuit-switched network where the effective topology and capacities can be adapted dynamically to changes in the traffic requirements or to changes in the physical network due to failure. Given the traffic requirements among all the O-D pairs, the objective is to reconfigure the physical network by designing a logical topology, assigning capacities to each logical link and determining the routing path for each O-D pair. In the circuit-switched network, the links are T1 lines. Each T1 line contains 24 DS0 channels, where each channel has a capacity of 64 Kbps. A logical link is a set-up circuit without relay at intermediate nodes. The capacity of a logical link can be described by the number of channels assigned. They assume that there is a queue associated with each channel and all queues operate independently and have infinite buffers. They adopt the average packet delay as the performance measure to be minimized, which is equivalent to minimizing the average number of packets in the network. Thus, they formulated the joint topology, capacity and routing problem as a nonlinear, mixed integer programming problem, i.e.,

$$\min \sum_{i \neq j} \frac{k_{ij} f_{ij}}{k_{ij} C - f_{ij}}, \tag{2.3.9}$$

where k_{ij} is the total number of channels used for logical links connecting nodes i and j, and f_{ij} is the aggregate rate of flows over logical links connecting nodes i and j. C is the capacity of a channel.

They first deal with the relaxation problem of (2.3.9) with the integer constraints relaxed. An important property is that the objective function is convex in k_{ij} and f_{ij}. This property implies that all traffic is routed over logical paths consisting of one logical link (direct VP), so that the routing problem is eliminated in the context. Thus they obtain a simplified relaxation problem (SRP), which can be solved by the reduced gradient method. Based on this solution to SRP, a LinkSelect and a RoundOff procedure are used to find the integer solution to the original problem. The LinkSelect procedure selects d logical links by applying a partial branch and bound method. The same strategy is used for reconfigurable ATM network [28]. The only difference is that total cell loss probability is instead the objective function to be minimized.

Logical Reconfiguration for Minimizing the Average Packet Delay

Gerla, Monteiro and Pazos [12] investigated the topology design of a packet switch network embedded into a backbone facility network. This is similar to the Lee and Yee's work [29]. The backbone (physical) network topology and capacities of physical links are given. They also assumed that the embedded network topology is given. The objective is to minimize the average packet delay over the routing variables and the capacities of logical links in the embedded network. It is assumed that the capacities of logical links are continuous variables, which is a major difference from [29]. Thus the optimization is formulated as

$$\min \frac{1}{\lambda} \sum_{u=1}^{\bar{M}} \frac{\bar{f}_u}{\bar{C}_u - \bar{f}_u}, \tag{2.3.10}$$

where $\bar{M}$ is the total number of logical links in the embedded network, $\bar{C}_u$ is the capacity of logical link u and $\bar{f}_u$ is the flow of logical link u.

The main difference from [29] is that the objective function (2.3.10) is not convex with respect to both $\bar{C}_u$ and $\bar{f}_u$. So the problem may have more than one local minimum. Gerla *et al.* [12] developed an algorithm for determining the capacities of logical links and routing. This algorithm tries to minimize the two linear functions over the capacities and routing respectively. The coefficients in the two functions are the first derivatives of (2.3.10) evaluated at the previous values of capacities and routing. The minimization over capacities is solved using the revised simplex method and the minimization over routing is solved by constructing a minimum path solution on the logical network. Then they use linear searching method such as the golden section technique to find a better solution. The above steps are repeated until no more improvement can be made.

Logical Configuration for Coexisting Virtual Subnetworks

Farago, Blaabjerg, Ast, Gordos and Henk [9] investigate the problem of optimizing virtual subnetworks (logical configuration) on a given backbone ATM network. They assume that a number of logical or virtual subnetworks can coexist, sharing the same physical transmission and switching capacities, on top of the physical infrastructure.

Let the backbone network have N nodes and K trunks (links). Denote by J the total number of virtual links over all virtual networks and let C_j be the capacity of the virtual link j. They assume that I traffic types (classes) are carried in the network. Thus in each virtual network there may be several O-D pairs, each with one type of traffic. The objective is to maximize the total expected network revenue W, i.e., the weighted version of the total expected carried traffic, given the physical network parameters and the traffic requirements of each virtual subnetwork. Let $\nu_{(v,p,i)}$ be the aggregated offered traffic rate of type i to O-D pair p in virtual network v. For a given virtual network v, O-D pair p and traffic type i, let $s^{(v,p,i)} = (s_{r_1}^{(v,p,i)}, s_{r_2}^{(v,p,i)}, \cdots)$ denote the load sharing vector, each component being the proportion that the load

is distributed among the routes that carry traffic type i for the O-D pair p. Thus the total network revenue W is defined

$$W = \sum_r w_r \nu_r (1 - L_r) = \sum_{v,p,i} \sum_{r \in R(v,p,i)} w_r \nu_{(v,p,i)} s_r^{(v,p,i)} (1 - L_r) \tag{2.3.11}$$

where L_r is the end-to-end blocking probability for traffic on route r, w_r is the weight on r and $R(v,p,i)$ is the set of the routes in a virtual network v realizing communication between the O-D pair p with traffic type i.

They introduce a two phase method to solve the above problem. In the first phase, the revenue W is approximated by a concave function. The concave function can then be maximized globally over the convex feasibility region by convex programming. In the second phase a gradient-based search is used to improve the solution to a refined model based on the initial point obtained in the first phase.

Logical VP-Based Network Design for Multi-Hour, Multi-Traffic Class ATM Networks

Medhi [32] considers a network where VCCs carrying similar traffic are statistically multiplexed onto VPCs, while VPCs are deterministically multiplexed into physical links. An optimization problem is formulated to minimize link capacity costs while guaranteeing the bandwidth requirements and QoS of multiple classes of traffic found for a given load period. Each source-destination pair only uses one direct VPC which should be selected among the candidate paths generated by the k-shortest path algorithm. The author presents a decomposition algorithm for the optimization problem using Lagrangian relaxation with duality and subgradient optimization. This decomposition algorithm results in an approximate solution to determine the capacity requirements and physical routes for all the VPCs in the resulting logical network.

Medhi [32] uses a complicated model to accurately compute or estimate the traffic requirements for different classes, so that the solution obtained from the optimization

problem is more realistic. However, including only the capacity costs of the VPCs in the objective function is not enough since the set-up costs of the VPCs are also significant for the performance of a logical VP-based network.

Medhi and Lu [33] investigate both the VP-based network design problem and the underlying transmission network dimensioning problem, taking into account the variation of traffic during the day in a heterogeneous multi-service environment for ATM networks. They first design a VP-based network topology for a given traffic demand pattern by assuming that a call connection may use a direct VPC or two hop VPCs for alternate routes. The VP-based network design problem is formulated as a mixed integer linear optimization problem where the objective is to minimize the total VPC cost. The effective bandwidths for different traffic classes are defined as the amount of bandwidth required to satisfy QoS and are calculated by different traffic models. Based on all the VPC capacities obtained from designing the VP-based network, the transmission network dimensioning problem determines the optimal bandwidth assignments for the physical links in the underlying transmission network. This is also formulated as a mixed integer linear program where the objective is to minimize the total link usage cost. The authors computationally compare the networks obtained by solving the two optimization problems with the networks under static-VP environment in terms of network cost. For various cost parameter sets, they found the cost savings to be as high as about 29% compared to a network with static-VP configuration.

A Layered Design Approach

Hui, Gursoy, Moayeri and Yates [19] introduce a layered switching architecture for integrated broadband networks, where switching at a layer is carried out to reduce the amount of blocking occurring at the next lower layer. In a broad sense, switching is the process of allocating transmission resources and the transfer of information along

one or more paths formed by the resources allocated. Blocking is the event that such allocation and transfer fail. A layered notion of equivalent bandwidth is introduced to facilitate the bandwidth management for heterogeneous service under the layered switching architecture. The equivalent bandwidth at each layer is calculated to satisfy the grade of service (GoS, here the blocking probability) at the above layer.

The problem of designing a VP network is formulated as the mathematical problem which minimizes path bandwidth cost subject to meeting GoS requirements. The path bandwidth cost is defined as the sum of the bandwidth costs used by the VPCs over all the links. They compare two VP layout set-up methods: the one where each VP is assigned a fixed bandwidth and the other where the bandwidth is shared by all the VPCs on a link.

A VP-Based Network Design Algorithm for ATM Networks

Ryu, Ohsaki, Murata, and Miyahara [42] consider the problem of designing a VP-based topology for ATM networks. The design problem is formulated as an optimization problem where the objective is to minimize the sum of the link cost and the node cost. The cost of each link is a linear function of the number of lines (each line has a fixed capacity). The cost of each node is proportional to the total bandwidth of all the VPs passing through the node. A heuristic algorithm is proposed to seek an approximate solution. The equivalent bandwidth for each demand pair is first calculated and a VP with the equivalent bandwidth is established for the demand pair using the shortest path as the initial solution. To reduce the network cost, the initial solution is improved by three heuristic methods: the alternation of VP route, the separation of a route into several routes for a VP, and the creation of VC switching nodes. They apply the design algorithm to two example networks to demonstrate how the design algorithm is used to configure the cost-effective VP-based network topology. The two VP-based networks are derived for two kinds of traffic respectively under each example.

2.4 Topology Design Methods for Survivable Networks

This section reviews some strategies for physical and logical topology design for survivable networks. An important protocol used in survivable networks is the distributed self-healing algorithm [15], which usually results in finding a set of successive shortest paths and reserving their maximum spare bandwidth to achieve fast restoration from a facility failure. The main difference between our design model and the following design methods for survivable networks, is that the extra capacity allocated in transmission links by our model is used for logical VP redesigning but not for physical fault tolerance only.

A Logical VP Reconfiguration Method for Survivable ATM Networks

Murakami and Kim [35] consider the problem of VP routing for survivable ATM networks. The objective is to find a VP configuration and bandwidth assignment in response to a dynamic change of network environment so that a self-healing algorithm can succeed. The concept of two step restoration is introduced to achieve fast restoration as well as optimal reconfiguration. They formulate the problem as a nonlinear, nonsmooth multicommodity flow problem with linear constraints. An algorithm is proposed to find a VP configuration and bandwidth assignment that minimizes the expected amount of lost flow upon restoration from a network failure. A modified flow deviation method is developed to obtain a near-optimal solution, where premature convergence to a nonsmooth point could be avoided by adjusting an optimization parameter. Finally they present the results of performance experiments are to show that the proposed routing scheme can detect the links that are vulnerable to a failure under the current traffic demand pattern and adjust a flow so as to improve the network survivability level.

Facility Planning and Circuit Routing for Survivable Networks

Nakamura and Oda [36] consider the VP level and the physical level planning problems for survivable networks. Here the transmission link failures may occur so that restoration capacity should be reserved. The VP level planning problem is a routing problem where optimal circuit routes are determined together with the allocation of spare capacity for assumed failure scenarios. They assume that each demand (source) pair should have multiple routes for survivability against link failures. The back-up capacity is allocated based on the restoration ratio for every circuit demand pair. The physical level planning problem is the facility (link) planning problem where the transmission facilities are added to minimize the installation cost while maintaining all the circuit demands. The facility planning problem also considers survivability against failures of the new facilities.

The two problems are formulated as integer programming problems. Nakamura and Oda [36] employ the Incremental Assignment (IA) method to solve the circuit routing problem. This method assigns a specified incremental unit of circuit demand for a selected route on a step-by-step basis in order to search for an optimal solution. The facility planning problem involves the circuit routing problem as a subproblem. They use the Genetic Algorithm (GA) method to search the solution space of facility plans and apply the IA method to evaluate each plan generated by the GA method.

A Comparison of Two Strategies for Survivable Network Design

Tsai, Coan, Kerner and Vecchi [46] compare a rapid reconfiguration approach with the conventional 1:1 protection switching approach using diverse routes at high speed networks. Two sample networks are employed for this comparison in terms of the installed first cost of the required network elements. The numerical results provide insight into the preferred range of applications for the two approaches although they show that no approach is consistently better than the other in terms of installed first costs.

2.5 Summary

In this chapter, an overview of existing work in physical and logical network designs has been given. The design problem for both physical and logical backbone networks usually consists of facility planning, capacity provisioning and commodity routing problems. Some papers only consider one of these three problems whereas others formulate a joint optimization problem for the topology design. The objective function to be minimized in most design problem formulations is the network cost or some network performance measure. Various optimization techniques are employed to solve the design problem, which usually result in tight link capacity assignments in order to match the expected traffic requirements and give the lowest cost. On the other hand, the purpose for survivable network design is to determine spare capacity and/or alternate routes in order to maximize the network's capability for physical fault tolerance. To react quickly to small variations in the network, capacity reallocation schemes are the best strategy for improving network efficiency. In this thesis we will provide a design model for physical ATM backbone networks with view to facilitate the resource management at the VP level and evaluate it using some VP bandwidth reallocation schemes.

Chapter 3

A Design Model for Physical Topology in ATM Networks

3.1 Introduction

In ATM networks, one important advantage of using VPs is to adapt to varying traffic and network failures through dynamic resource management such as VP reconfiguration and dynamic capacity reallocation. The extent to which a dynamic resource management scheme is able to provide the adaptability to changing traffic conditions and prevent possible network failures is heavily dependent on the physical topology of underlying ATM network. In this chapter, we consider the physical topology design for local or wide-area ATM networks with a view to facilitating the dynamic resource management at the VP level.

We consider the design of the backbone network and assume that the local access networks have been designed. The centralized access networks attached to each backbone node can be designed by various techniques in [21]. We also assume that the locations of all backbone switches are selected by network designers or managers, or determined by some clustering methods.

We first introduce the design model in Section 3.2. This physical topology design

is formulated as a nonlinear mixed integer programming problem. The objective cost function to be minimized consists of network set-up cost and the capacity usage cost. Two parameters are introduced in the optimization problem to make a trade-off between the cost and certain topological properties which are required to provide higher flexibility for the dynamic resource management at VP level. In Section 3.3 we propose a graph-based heuristic algorithm. An alternative algorithm is given in Section 3.4 for a sparse network. The graph theory used for deriving the complexity of these two algorithms is presented in the last two sections.

3.2 Problem Formulation

An ATM network should accommodate various services for multimedia traffic with different qualities of services (QoSs) in an integrated environment. The logical VP reconfiguration and dynamic capacity reallocation provide an effective mechanism for ATM networks to dynamically adapt to changing traffic conditions and possible network failures. The physical topology of an ATM network should be designed to provide greater flexibility to the logical reconfiguration mechanism.

In the context of this thesis, the physical topology design for the ATM backbone network consists of deciding on the set of transmission links which interconnect the backbone switch nodes, the capacity of these links and the routes used by traffic between all source pairs. The objective for our ATM network design is to minimize the sum of total link capacity costs and total link set-up costs. The capacity usage cost for each transmission link is the operation cost for managing the amount of traffic transmitted on the link. The link set-up cost is the link installation cost which may include construction and purchasing fees for fiber-optic cables.

Besides low cost, another requirement is that the design be robust with respect to changing traffic conditions and possible network failures. Using the VP concept,

we can construct a logical VP-based network to facilitate the traffic management for current traffic demands. The logical network can be reconfigured to handle the fluctuation in traffic demands. The physical network design can help the ability of the logical network to adapt in two ways:

- The physical network should provide at least two disjoint routes for each switch node pair. This gives the routing algorithm more options for routing calls without needing to change the routes of the VPs, which is an expensive operation [10]. Providing more routes also helps the implementation of the restoration algorithms against network facility failures.

- Each transmission link should have some spare capacity. When the traffic demands change, spare capacity is required for performing a fast bandwidth reallocation algorithm rather than running a time-consuming VP reconfiguration based on some optimization model.

In our design model, we assume that the network designer has found a set of switch locations for the backbone network. Formation of the location set might be based on practical considerations or might be produced by some clustering algorithm. Let n be the number of switch locations for the ATM backbone network. For a node pair (i, j), $i \neq j$, we use a_{ij} to represent the capacity cost per unit if a transmission link L_{ij} is established between the pair in the physical network. Intuitively this depends on the geographical distance of the link L_{ij}. The term b_{ij} is used to denote the initial set-up cost for the link L_{ij}. It is mainly related to the transmission trunk prices and installation charges which are also some functions of the physical distance of the link L_{ij}. The fixed cost b_{ij} may also be related to the traffic amount on the link L_{ij}. We assume that there exists no link within a switch node so that $a_{ii} = b_{ii} = 0$ for $i = 1, \cdots, n$. To simplify the design model, we exclude all possible costs occurring at

the switch nodes, that is, the switch nodes can handle any amount of traffic and can be placed without any cost.

The external traffic characterization is an important input to the final network topology. We assume that the external traffic requirements are given or may be estimated from previous experience. The required bandwidths can be determined by equivalent capacity methods [16] [32]. Because all connections are statistically multiplexed at the physical layer and the bit rate of connections varies in ATM networks, it is important to characterize, for a given quality of service (QoS), the effective bandwidth requirement of both individual connections and the aggregate bandwidth usage of connections multiplexed on a given link. The methods in [16] and [32] provide computationally simple approximations for the equivalent capacity or bandwidth requirement of a single or multiplexed connections on the basis of their statistical characteristics. We denote by r_{ij} the traffic demand between the node pair (i,j), $i \neq j, i,j = 1,\cdots,n$. r_{ij} can represent the total multiplexing or accumulated bandwidth required for multiple classes of traffic between a node pair. Since the traffic in the network changes periodically, r_{ij} should be calculated as an average over a long time interval to make the physical network robust. It is assumed that $r_{ii} = 0$, that is, there is no traffic flow within a node.

For the convenience of the reader, the notation introduced in this section is summarized in Table 3.1.

Now we look at the optimization problem. Let N be the set of all the switch nodes, that is, $N = \{1,\cdots,n\}$. The physical topology design problem is to minimize the objective cost function:

$$\eta = \sum_i \sum_j \{a_{ij}c_{ij} + b_{ij}I(c_{ij} > 0)\}, \tag{3.2.1}$$

subject to:

$$c_{ij} = (1+\rho) \sum_{(k,h)\in P_{ij}^{(1)}} r_{kh} + \alpha \sum_{(k,h)\in P_{ij}^{(2)}} r_{kh}$$

Table 3.1: Notation

N	the set of all switch nodes, $N = \{1, \cdots, n\}$
S	the set of created transmission links (i.e., having positive capacity)
L_{ij}	the high speed transmission link between switch nodes i and j.
d_{ij}	the physical distance between two nodes i and j
a_{ij}	the link cost per unit bandwidth used in the link L_{ij}
b_{ij}	the set-up cost for establishing the link L_{ij}
r_{kh}	the traffic demand between the source pair (k, h)
c_{ij}	the total capacity assigned to the link L_{ij}
$P_{ij}^{(1)}$	the set of all source pairs using the link L_{ij} in the primary route
$P_{ij}^{(2)}$	the set of all source pairs using the link L_{ij} in the secondary route
x_{ij}^{kh}	a 0-1 variable indicating whether the primary route for the source pair (k, h) traverses the link L_{ij} or not
y_{ij}^{kh}	a 0-1 variable indicating whether the secondary route for the source pair (k, h) traverses the link L_{ij} or not
ρ	the parameter to control the extra capacity on the primary route
α	the parameter to control the capacity on the secondary route

$$= (1+\rho)\sum_{k}\sum_{h} x_{ij}^{kh} r_{kh} + \alpha \sum_{k}\sum_{h} y_{ij}^{kh} r_{kh}, \forall i, j \in N, i \neq j, \quad (3.2.2)$$

$$x_{ij}^{kh}, y_{ij}^{kh} \in \{0,1\},\ \forall i, j \in N, i \neq j,\ \forall k, h \in N, k \neq h, \quad (3.2.3)$$

$$x_{ij}^{kh} + y_{i'j'}^{kh} \leq 1,\ \text{for } i = i' \text{or } j = j', \forall k, h \in N, k \neq h, \quad (3.2.4)$$

where c_{ij} is the total capacity assigned to the link, L_{ij}. Note that the link capacities are continuous rather than discrete. Here $I(c_{ij} > 0)$ is the indicator function of the set $\{c_{ij} > 0\}$, that is, it is 1 if $c_{ij} > 0$, otherwise 0. $P_{ij}^{(1)}$ is the set of all the source pairs using the link L_{ij} in the primary route, and $P_{ij}^{(2)}$ is the set of all the source pairs using the link L_{ij} in the secondary (back up) route. The constraint (3.2.3) means that x_{ij}^{kh} and y_{ij}^{kh} are 0-1 variables. $x_{ij}^{kh} = 1$ implies that the primary route for the node pair (k, h) traverses the link L_{ij}. $y_{ij}^{kh} = 1$ indicates that the secondary route for the node pair (k, h) traverses the link L_{ij}. The constraint (3.2.4) forces the primary and secondary routes for each source pair to be node-disjoint.

In the above design problem, the topology of physical network and link capacity assignment are determined by the values of x_{ij}^{kh}, y_{ij}^{kh} and c_{ij}. The two parameters, ρ and α, are included in the model (3.2.1) in order to control the amount of spare capacity assigned to the transmission links and the amount of bandwidth allocated to the alternate paths respectively. Consequently, the denseness of the topology will be affected by the two parameters because the capacity cost in the objective function is a function of the amount of link capacity. Roughly speaking, larger values of ρ and α result in more spare capacity allocated on the links. The effect of these two parameters on the physical topology generated based on the model (3.2.1)-(3.2.4) will be explored in Chapter 4.

The objective function (3.2.1) is composed of two types of costs. If the objective contains only the link set-up costs, a designer prefers selecting the link with small set-up cost, which usually results in connecting only neighboring nodes and consequently in a sparse network. On the other hand, if the objective includes only the link

capacity costs, the designer must route each traffic demand through the shortest paths with respect to the link capacity cost, which typically results in a very dense network. We include both the link capacity costs and the link set-up costs in the objective function (3.2.1) to provide a compromise between these two extreme cases. In general, a transmission link should be established between two switch nodes if the link set-up cost is smaller than the cost for routing the traffic between the two switches on any other links.

The following theorem gives a characterization of the optimal solution to the design model (3.2.1)-(3.2.4), which claims that if the physical topology is known *priori*, the first two node-disjoint shortest paths provide the optimal routing for each source pair. Note that in this chapter the shortest path is defined using the link capacity cost, a_{ij}, as the link distance.

Theorem 3.2.1 *Given ρ, α and the set of transmission links (i.e., $\{(i,j)|c_{ij} > 0\}$), the optimal topology can be determined by choosing the first two node-disjoint shortest paths between each source pair as the primary and secondary routes.*

Proof: When the set of transmission links $S = \{(i,j)|c_{ij} > 0\}$ is given, the problem (3.2.1)-(3.2.4) is reduced to determining the link capacity assignments which minimize the following cost function:

$$\eta_1 = \sum_{(i,j)\in S} a_{ij}c_{ij} \tag{3.2.5}$$

subject to (3.2.2)-(3.2.4). Substituting (3.2.2) into (3.2.5), we obtain the total cost

$$\begin{aligned}\eta_1 &= \sum_k \sum_h r_{kh}\{(1+\rho)\sum_{(i,j)\in S} a_{ij}x_{ij}^{kh} + \alpha \sum_{(i,j)\in S} a_{ij}y_{ij}^{kh}\} \\ &= \sum_k \sum_h r_{kh}\{(1+\rho)\sum_{(i,j)\in P_{kh}^{(1)}(S)} a_{ij} + \alpha \sum_{(i,j)\in P_{kh}^{(2)}(S)} a_{ij}\}\end{aligned} \tag{3.2.6}$$

where $P_{kh}^{(1)}(S)$ and $P_{kh}^{(2)}(S)$ are two node-disjoint paths between the source pair (k,h) in S. Since the r_{kh} are fixed, (3.2.6) is minimized if $P_{kh}^{(1)}(S)$ and $P_{kh}^{(2)}(S)$ are chosen

to be the first two node-disjoint shortest paths between the pair (k,h) in S. This completes the proof.

■

The above characterization will be used in the next section to develop heuristic algorithms to find a local minimum to the original problem.

3.3 A Graph-Based Heuristic Algorithm

The network design problem given by (3.2.1)-(3.2.4) is a nonlinear, mixed integer programming problem which is NP-hard [11]. In this section, we develop a heuristic graph algorithm with a polynomial running time.

According to the design model in Section 3.2, finding the optimal physical topology corresponds to determining the placement of links and the assignment of link capacity in order to minimize the objective cost (3.2.1). The design model (3.2.1)-(3.2.4) requires that the optimal physical topology provides two node-disjoint paths between each source pair. Theorem 3.2.1 tells us that, if we have selected the set of transmission links, then we must choose the first shortest path as the primary route and the second shortest path as the secondary route for each pair in order to minimize the cost objective function. Another important feature of the model is that the topology will have the property of 2-connectivity (see the definition in Section 3.5) since two node-disjoint routes are provided for each source pair. The two features are incorporated in the following heuristic algorithm to solve the optimization problem.

In general, the algorithm consists of 3 phases: initialization, link deletion and link addition. In the initialization phase, a complete graph on the n backbone nodes is formed. At each iteration of the next phase, the Link Deletion Phase, the link whose removal reduces the cost the most while still maintaining the 2-connected property is deleted. This phase ends when there is no link that will futher reduce the cost.

Then, in the Link Addition Phase, at most k links are added back into the topology (those that give the most cost reduction). At each iteration of Link Deletion and Link Addition, the primary and node-disjoint (with respect to the primary route) secondary routes are found between each source pair in order to assign capacity and calculate the total cost. These routes are based on shortest distance paths.

We present an outline of the algorithm and a discussion of its complexity as follows:

1. Initialization Phase:
   - Form the complete graph on n nodes
   - Find primary and node-disjoint secondary routes
2. Link Deletion Phase: Iterate while a cost improvement can be made with a link deletion.
   - Step 1. Consider each link in turn
     - a) Temporarily delete link
     - b) Check 2-connectivity. Go to the next link if this property is violated
     - c) Find primary and node-disjoint secondary routes for each source pair using shortest paths
     - d) Calculate cost improvement
     - e) Add link back to topology
   - Step 2. Delete the link that produced the largest cost improvement in Step 1.
   - Repeat the Link Deletion Phase with the new graph.
3. Link Addition Phase: Find at most k links to add back to the network
   - Step 1. Consider each missing link in turn

- a) Temporarily add link
- b) Find primary and node-disjoint secondary routes for each source pair using shortest paths
- c) Calculate cost improvement
- d) Delete link

- Step 2. Add the link that produced the largest cost improvement in Step 1.

- Repeat the Link Addition Phase with the new graph.

Once the complete graph is created, the first shortest path between all source pairs is found using Floyd's algorithm [21], which is $O(n^3)$. The second shortest path is found for each source pair by deleting the edges and nodes in the first shortest path and applying Dijkstra's shortest path algorithm. Thus, finding all the second shortest paths requires $O(n^2)$ applications of Dijkstra's algorithm. Since Dijkstra's algorithm has running time $O(n^2)$, the complexity of finding all the second shortest paths is $O(n^4)$. Therefore, the complexity of the Initialization phase is $O(n^4)$.

In the Link Deletion Phase, at most $O(n^2)$ edges are actually deleted to create a new topology. Thus, at most $O(n^2)$ iterations can be performed in this phase. For each new topology, we need to calculate all the reduced costs by deleting $O(n^2)$ edges in order to find the best edge to delete. To evaluate the reduced cost, the feasibility of deleting an edge (that is, maintaining 2-connectivity) and the placement of capacity (finding primary and secondary routes) must be determined.

The complexity of checking the 2-connectivity can be reduced by exploiting the iterative nature of the Link Deletion Phase. Note that the number of links is reduced by one for each iteration. The complexity is reduced from $O(n^3)$ to $O(tn^2)$ where t is the number of nodes in the shortest path between the two end nodes of the deleted

link. This is proved and described more fully in Section 3.5. Although t may be $O(n)$ in the worst case, in practice it tends to be much smaller.

Similarly, we can exploit the iterative nature of the Link Deletion Phase to reduce the complexity of finding the primary and secondary paths. We use the method of Murchland [21] to modify the first shortest path between all pairs of switch nodes, so that the recalculation for the first shortest path between all pairs, given the removal of an link, takes $O(n^2)$ in the worst case. The complexity is usually considerably lower. We also extend the Murchland method to modify the second shortest path, so that the recalculation for all source pairs takes $O(qn^2)$, where q is the number of first and second shortest paths passing through the deleted link. Typically, q depends on how dense the network is and what the average hop counts of the two routes are. In the worst case, q may be $O(n^2)$. But in practice it is much smaller and is usually bounded by $O(n)$ in a 2-connected graph. Thus, the complexity for modifying all the first and second shortest paths is $O((1+q)n^2)$. Since each iteration needs to try deleting all possible links, then it takes time $O((1+t+q)n^4)$ (at worst case $O(n^6)$) to create a new topology. Therefore, the complexity for the entire Link Deletion Phase is $O(n^8)$ at the worst case since at most $O(n^2)$ actual link deletions can be made. But the actual complexity is much lower as can be seen in computational experiments of Chapter 4 where q is quite small.

For the Link Addition Phase, the 2-connectivity does not have to be checked since it is inherited when more edges are added. However, the complexity improvements in finding the primary and secondary routes in the Link Deletion Phase do not work in the Link Addition Phase and the complexity of each addition can be as high as $O(n^6)$. To combat this, the number of iterations is limited to a constant k and, in fact, in practice the number of links that can be added back and still reduce cost is small ($k = 2$ is adequate for the examples in this thesis). This means that the whole heuristic algorithm has complexity of $O(n^8)$ at worst case but usually has complexity of $O(n^6)$.

Theoretically, after the Link Addition Phase we may delete some link to further reduce the cost. That is, the Link Deletion Phase and the Link Addition Phase should be repeated until no cost improvement can be made. However, computational experiments show that in most cases repeating the two phases one more time results in little cost improvement. For computational efficiency, we only include the first round of the Link Deletion Phase and the Link Addition Phase in this algorithm.

3.4 An Alternative Heuristic Algorithm

In the last section, the graph-based heuristic algorithm started with the complete graph as the initial topology. If the optimal physical topology is very sparse, then this algorithm needs a very large number of iterations of link deletions to reach a local minimum. In this section, we give an alternative heuristic algorithm which starts with a minimal 2-connected graph. If the minimal 2-connected graph is chosen to be close to the optimal physical topology (in the sense that they do not have many different links), we can expect the alternative heuristic algorithm will take less time to find a local minimum. In the remainder of the thesis, we will call the algorithm in the last section Algorithm 1 and this alternative algorithm Algorithm 2. In Algorithm 2, a local search method from [40] is applied to find a minimal 2-connected graph in the Initialization Phase. Then we go to the Link Addition Phase and the Link Deletion Phase to improve the objective cost.

The local search algorithm [40] for creating a 2-connected directed graph is described as follows. In the beginning, all the nodes are isolated, that is, all the nodes have out-degrees 0 and in-degrees 0. At each iteration, we add an edge (link) between a node with smallest out-degree and one with smallest in-degree. If there are more nodes with smallest out-degree and/or smallest in-degree, we choose the link with smallest set-up cost. At each iteration, 2-connectivity is checked and the search stops

when a 2-connected graph is found. The first 2-connected graph is the initial topology for the improvement phase.

The alternative heuristic algorithm is outlined as follows:

1. Initialization Phase: Iterative local search.
   - Step 1. Start with the isolated graph
   - Step 2. Find the nodes with smallest out-degree and those with smallest in-degree in the current network.
     - a) Among the links between the nodes with smallest out-degree and those with smallest in-degree, choose the one with smallest set-up cost and add it to the network.
     - b) test if the network is 2-connected or not.
   - Repeat Step 2 until a 2-connected network is found.
2. Link Addition Phase: Iterate while a cost improvement can be made with a link addition.
3. Link Deletion Phase: Iterate while a cost improvement can be made with a link deletion.

In the Initialization Phase, adding a link to the network takes time of $O(n^2)$ since finding the nodes with smallest out-degree needs n comparisons and choosing the link with smallest set-up cost needs at most $O(n^2)$ comparisons. Testing 2-connectivity may take time of at most $O(n^3)$. We can add at most $O(n^2)$ links in the Initialization Phase. So the worst running time of this phase is $O(n^5)$. Since most early iterations do not produce a 2-connected network, the test of 2-connectivity can be replaced by checking the degrees of all the nodes or by checking the connectivity. Usually the first 2-connected network is very sparse (it is minimal), so that the complexity of this phase is actually much less than $O(n^5)$.

The bottleneck of the algorithm is in the Link Addition Phase. As shown in the last section, each link addition in this phase takes time of $O(n^6)$. In the worst case, we may need $O(n^2)$ link additions. Therefore, the worst running time of the heuristic algorithm is $O(n^8)$. Computational experiments to show the difference between Algorithms 1 and 2 in running time are conducted in the next chapter.

The two heuristic algorithms were implemented in C and run on SUN Unix machines. For moderate sizes of networks (below 30 nodes), it takes affordable time. For large networks, the heuristics could be applied hierarchically using a clustering mechanism. To show how the algorithms produce the physical topology with flexibility for logical reconfiguration and dynamic capacity allocation, we evaluate the physical networks generated by the two heuristic algorithms statically in terms of link utilization, network connectivity and average route hops, and dynamically in terms of average call-blocking probability using several example networks in the next chapter.

3.5 Two-Connectivity for Link Deletion

In the previous two sections, each iteration in the Link Deletion Phase of the two heuristic algorithms maintains the property of 2-connectivity. In this section, we find an efficient method to check for the 2-connectivity given the deletion of one link for a 2-connected graph. First we define the 2-connectivity of an undirected graph.

Definition 3.5.1 *A vertex cut of a graph G is a set $S \subseteq V(G)$ such that $G - S$ has more than one component. A graph G is 2-connected if every vertex cut has at least 2 vertices.*

Checking the 2-connectivity of a graph G is shown to be equivalent to finding two node-disjoint paths between each pair of nodes by the famous Whitney theorem [47].

Theorem 3.5.2 *(Whitney 1932) An undirected graph G having at least three vertices is 2-connected if and only if each pair $u, v \in V(G)$ is connected by a pair of node-disjoint u, v-paths in G.*

The following theorem gives a more efficient way to check the 2-connectivity of a graph G than finding two node-disjoint paths. Its proof can be found in [47] and is omitted here.

Theorem 3.5.3 *An undirected graph G having at least three vertices is 2-connected if and only if G is connected and has no cut-vertex.*

We can use the depth first search [21], which has complexity of $O(n^2)$, to check if a node is cut or not. Thus, checking the 2-connectivity of a graph G has complexity of $O(n^3)$. The iterative nature of the Link Deletion Phase can be exploited to reduce the cost for checking 2-connectivity when exactly one link is deleted at a time. The 2-connectivity of the reduced graph can be characterized as follows:

Theorem 3.5.4 *Let G be a 2-connected undirected graph and e be an edge between u and $v \in V(G)$. Then $G - \{e\}$ is 2-connected if and only if there are two node disjoint paths between u and v in $G - \{e\}$.*

Proof: The necessity is easily obtained by Whitney theorem.

To show the sufficiency, we need to prove that every pair (x, y) has two node-disjoint paths in $G - \{e\}$. Note that x and y refer to node not the 0-1 variables in the design model. Let C denote the cycle formed by the two node-disjoint paths between u and v in $G - \{e\}$. Consider an arbitrary pair of nodes x and y. Since G is 2-connected, there exist two node-disjoint paths P and Q between x and y in G. To construct two node-disjoint paths for pair (x, y) in $G - \{e\}$, we have the following two cases to consider:

Case 1. If P and Q do not use link e, then these two paths are node-disjoint in $G - \{e\}$.

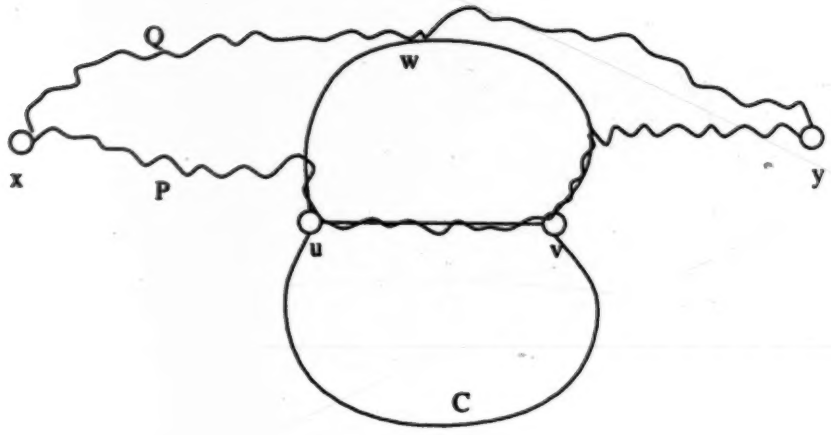


Figure 3.1: P passes through link e and Q touches C only in vertex w

Case 2. If one of P and Q uses the link e, we may assume that P contains e without loss of generality since P and Q are node-disjoint. One of the following three situations holds:

(1) If Q doesn't pass through any vertex in the cycle C, then we can reroute the path P by replacing the link e of P by either of the two paths between u and v in $G - \{e\}$. We denote the new path by P'. Thus, we obtain two node-disjoint paths P' and Q for the pair (x, y) in $G - \{e\}$.

(2) If Q passes through only a vertex w in the cycle C, then we can reroute the path P by replacing the link e of P by the u, v path that does not contain w in $G-\{e\}$; see Figure 3.1. We denote the new path by P'. Thus, we obtain two node-disjoint paths P' and Q for pair (x, y) in $G - \{e\}$.

(3) If Q passes through more than one vertex in the cycle C, then we can find two vertices x_q and y_q in path Q, where x_q and y_q are the first vertices intercepting C from x and from y respectively. Also we can find two vertices x_p and y_p in path P, which are the first vertices intercepting C from x and from y respectively. See Figure 3.2 for an illustration. We construct path P' by starting from x, traversing P until x_p,

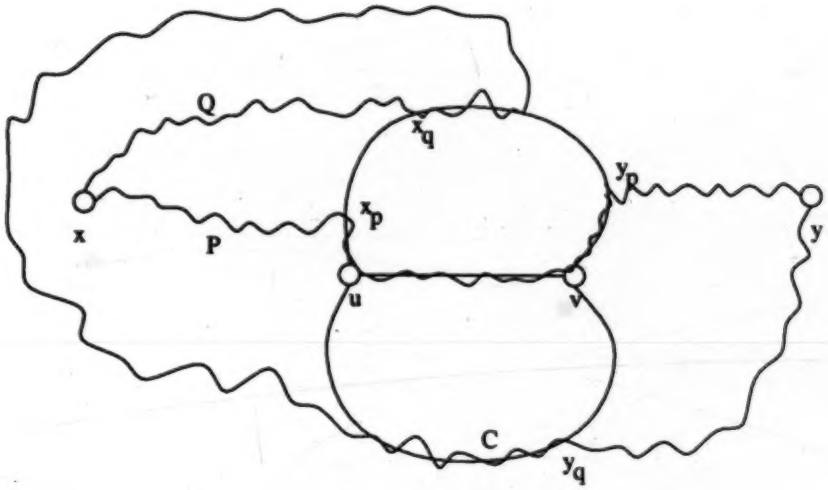


Figure 3.2: P passes through link e and Q has several intersections with cycle C

then traversing the cycle C in the opposite direction from x_q until meeting the first one of y_p and y_q. Then, if y_p was reached first, P is followed to reach Y, or if y_q was reached first, Q is followed to reach Y. Similarly, we construct path Q' by starting from x, going to x_q over Q and then traversing C away from x_p until reaching y_p if y_q was reached first for P', or y_q if y_p was reached first for P'. Finally Q' traverses over P or Q to reach y. Thus, we obtain the following possible routing sequences for P' and Q': (a) P': $x->P->x_p->C->y_q->Q->y$, Q': $x->Q->x_q->C->y_p->P->y$, or (b) P': $x->P->x_p->C->y_p->P->y$, Q': $x->Q->x_q->C->y_q->Q->y$. Since P' and Q' use disjoint parts of the cycle C and P and Q are disjoint, then P' and Q' are node-disjoint in $G-\{e\}$.

■

Theorem 3.5.4 results in an efficient way to check the 2-connectivity for the reduced graph.

Theorem 3.5.5 *Let G be a 2-connected undirected graph and e be an edge between u and $v \in V(G)$. Then $G - \{e\}$ is 2-connected if and only if any node in the shortest path between u and v in $G - \{e\}$ is not a cut-vertex.*

Proof: By theorem 3.5.4, we need to prove that two node-disjoint paths in $G - \{e\}$ exist between u and v if and only if any node in the shortest path between u and v in $G - \{e\}$ is not cut-vertex. The above statement is equivalent to: u and v have only one path if and only if there exists a cut-vertex in the shortest path between u and v in $G - \{e\}$.

Necessity: if u and v have only one path then by Menger theorem [47] there exists a vertex in $G - \{e\}$ whose deletion breaks the only path between u and v. This cut-vertex must be in the only path which is the shortest path.

Sufficiency: Since there exists a cut-vertex, by Menger theorem we know that there is at most one path between u and v. But, there exists a shortest path linking u and v because $G - \{e\}$ is connected. Hence this shortest path is the only path between u and v.

■

Using Theorem 3.5.5, we can check whether all the nodes in the shortest path between u and v in $G - \{e\}$ are cut-vertices or not in order to test the 2-connectivity of $G - \{e\}$. If t is the number of nodes in the shortest path between u and v, the complexity of testing the 2-connectivity of $G - \{e\}$ is $O(tn^2)$.

3.6 Rerouting for Link Deletion

In this section, we apply Murchland's method [21] to modify the first shortest path between all pairs of switch nodes after a link is deleted. We also extend the Murchland method to modify all the second shortest paths when deleting a link in order to reduce the running time.

Let G be a directed graph with n nodes indexed by $1, \cdots, n$. Denote by $dist(i,j)$ the length of the link from nodes i to j. If there exists no link from nodes i to j, then $dist(i,j)$ is set to be infinite. Let $sp_dist(i,j)$ and $pred(i,j)$ denote the length and routing of the shortest path from nodes i to j, respectively. $pred(i,j)$ gives the next to last node in the shortest path from i to j and can be used to backtrack the path from i to j. We assume that the two matrices *sp_dist* and *pred* have been obtained using Floyd's algorithm based on the length matrix *dist*. We know that the complexity of Floyd's algorithm is $O(n^3)$. If a link (k,h) is removed from the graph G, we can recalculate two matrices *sp_dist* and *pred* more efficiently using Murchland's method. Murchland's method is based on the following important fact. If the link (k,h) was not part of the shortest path from i to j before it was removed, it is certainly not part of the path afterwards. So it is only necessary to check pairs (i,j), whose shortest path lengths satisfy:

$$sp_dist(i,j) = sp_dist(i,k) + dist(k,h) + sp_dist(h,j)$$

Note that if $dist(k,h)$ is not part of the shortest path from k to h, then no paths change at all.

Murchland's method is described in following pseudo-code:

```
Murchland (n, k, h, dist, sp_dist, pred)
        pairs: link list of node pairs
        If (dist(k,h) > sp_dist(k,h))
          return (sp_dist, pred)
        pairs = NULL
        For each i from 1 to n
          For each j from 1 to n
            If (sp_dist(i,j) = sp_dist(i,k) + dist(k,h) + sp_dist(h,j))
              append ((i,j), pairs)
```

$sp_dist(i,j) = Infinite$

$dist(i,j) = Infinite$

For each l from 1 to n

For each (i,j) in *pairs*

If $(sp_dist(i,j) > sp_dist(i,l) + sp_dist(l,j))$

$sp_dist(i,j) = sp_dist(i,l) + sp_dist(l,j)$

return $(sp_dist, pred)$

The complexity of Murchland's method is $O(pn)$, where p is the number of node pairs in the set *pairs*. In the worst case, it is possible for *pairs* to contain $O(n^2)$ node pairs and for the complexity to be $O(n^3)$, like Floyd's algorithm. In practice, however, p tends to be much smaller in most cases.

Using the same idea as Murchland's method, we can use the following procedure to efficiently modify the second shortest path between all node pairs. To recalculate the second shortest paths, we only need to consider those pairs whose first shortest or second shortest path passed through the link (k,h) before it is removed. Let $second_dist(i,j)$ and $second_pred(i,j,\cdot)$ denote the length and routing vector of the second shortest path between i and j respectively. We list the pseudo-code for this procedure as follows.

$Modify_Second_Shortest_Path(n,k,h,dist,sp_dist,pred,second_dist,second_pred)$

pairs: link list of node pairs whose shortest path uses the link (k,h)

pairs_2: link list of node pairs whose second shortest path uses (k,h)

Create *pairs* in the same way as the procedure *Murchland*

Search the matrix *second_pred* to find all pairs whose second shortest path uses the link (k,h) and put them into *pairs_2*

For each pair (i,j) in either *pairs* or *pairs_2*

Temporarilly delete all the nodes and links in the modified shortest path

from i to j

Call Dijkstra algorithm to obtain the new *second_dist*(i,j)

and *second_pred*(i,j)

return (*second_dist*, *second_pred*)

The complexity of this procedure is $O(qn^2)$, where q is the number of node pairs in the sets *pairs* and *pairs_2*. In the worst case, it could be $O(n^4)$ since it is possible for q to be $O(n^2)$. But, q is usually quite small in a 2-connected graph G and in practice it tends to be much smaller.

Chapter 4

Experiments for Performance Evaluation

Our computational experiments focus on two primary issues. First, we make a comparison between the two heuristic algorithms with respect to the computational efficiency and the quality of solutions. Second, we investigate certain graph properties of the physical networks generated by one of two heuristic algorithms, which are required to support the reconfigurability of the network. We also conduct a call-level simulation study to evaluate how the parameters of the physical topology design model affect the performance of a dynamic bandwidth reallocation scheme in the VP reconfiguration phase.

4.1 Comparison for Two Algorithms

In this section, we apply the two heuristic algorithms to several sample networks to show their effectiveness. The physical topologies generated by the two heuristic algorithms are local minima under the design model (3.2.1)-(3.2.4). The two heuristic algorithms are compared with regard to running time, total network cost and denseness of network topology.

To generate a physical ATM network, we should first select the locations of all switch nodes and know the long term traffic requirements of all possible pairs. The node locations used in the computational experiments come from the following sample networks: the metropolitan area network (METRO) [42] with 8 nodes whose locations constitute an equal-edge octagon, the NSFNET T3 backbone network (NSF) [42] with 12 nodes, and three sample networks which are extracted from an actual voice network spanning the continental US [32]. The three sample networks from [32] are denoted as EN-1 of 10 nodes, EN-2 of 18 nodes, and EN-3 of 23 nodes.

The link capacity cost a_{ij} and the link set-up cost b_{ij} are determined by the network designer. In our numerical experiments, we set both a_{ij} and b_{ij} to be equal to the physical distance of the switch pair (i,j), $i,j = 1,\cdots,n$.

The traffic requirement for each source pair is given by the equivalent bandwidth in order to maintain the required QoS. We consider normalized uniform and non-uniform traffic requirements for comparison purposes. For uniform traffic, we let $r_{kh} = 1$ for $k \neq h$, $k,h = 1,\cdots,n$. The non-uniform traffic is normalized such that $\sum_{k\neq h}^{n} r_{kh} = n(n-1)$ holds, so that, the total amount of traffic is equal to that for the uniform case. The traffic requirements for the non-uniform case are taken from [42] with minor modifications.

In this section, we only present the computational results for EN-1 under uniform traffic since experiments for other sample networks and non-uniform traffic produce similar results. We use each algorithm to generate 16 physical topologies for $\rho = 0.0, 0.1, 0.3, 0.5$ and $\alpha = 0.0, 0.1, 0.3, 0.5$. For each topology, we calculate the CPU time spent, corresponding network cost, and link percentage which is defined as the fraction of established transmission links over all possible links.

The distances between all the pairs of switch nodes in EN-1 network are listed in Table 4.1. The results for EN-1 under uniform traffic are displayed in Figures (4.1)-(4.3).

Table 4.1: The distance data for EN1

	1	2	3	4	5	6	7	8	9	10
1	0	14.58	13.05	16.82	21.25	28.15	35.38	37.17	45.07	40.70
2	14.58	0	11.25	18.75	26.37	17.35	31.55	30.10	34.00	27.50
3	13.05	11.25	0	7.50	15.21	16.09	23.05	24.14	32.52	29.71
4	16.82	18.75	7.50	0	7.91	19.27	19.04	22.53	33.66	33.28
5	21.25	26.37	15.21	7.91	0	26.10	20.00	25.77	38.59	39.86
6	28.15	17.35	16.09	19.27	26.10	0	18.84	14.25	17.03	14.02
7	35.38	31.55	23.05	19.04	20.00	18.84	0	8.00	22.11	28.09
8	37.17	30.10	24.14	22.53	25.77	14.25	8.00	0	14.14	20.62
9	45.07	34.00	32.52	33.66	38.59	17.03	22.11	14.14	0	11.18
10	40.70	27.50	29.71	33.28	39.86	14.02	28.09	20.62	11.18	0

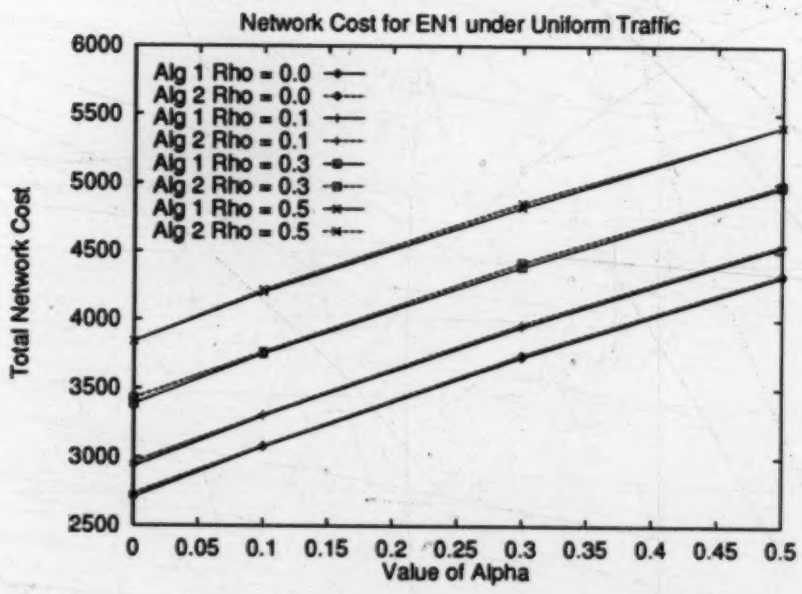


Figure 4.1: The Comparison of Network Cost between the Two Algorithms

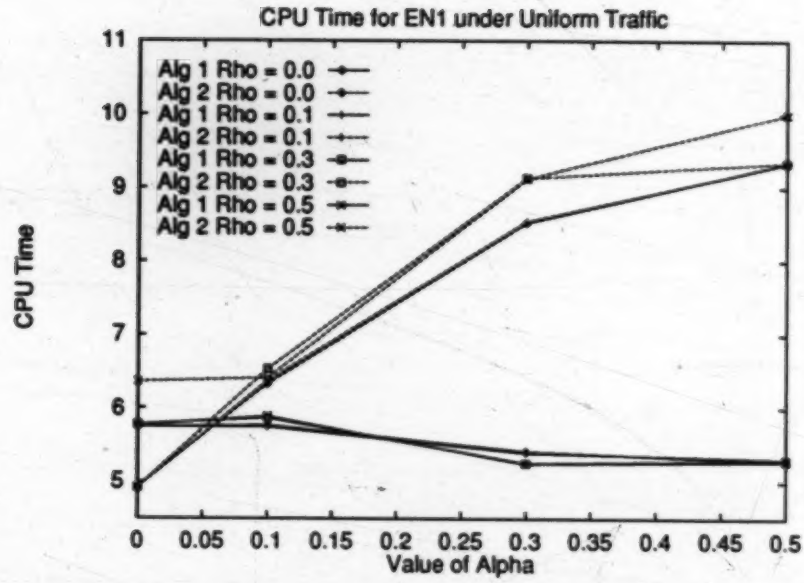


Figure 4.2: The Comparison of CPU Time between the Two Algorithms

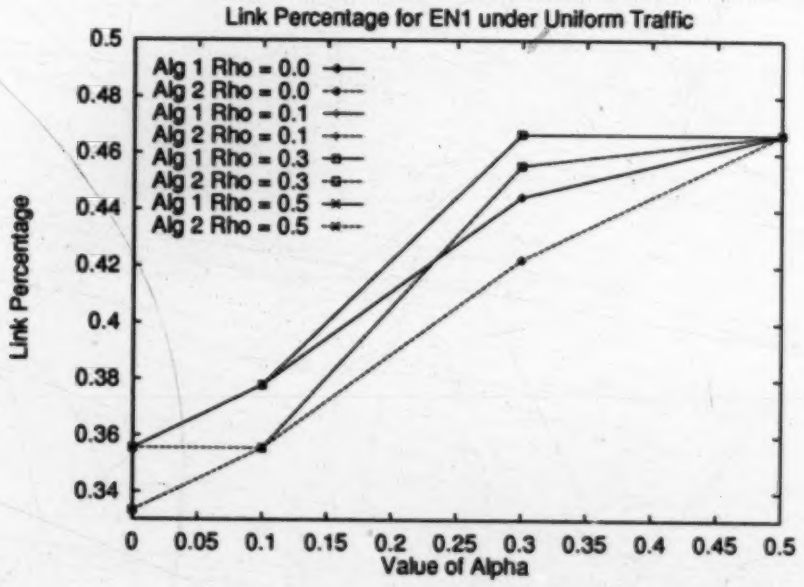
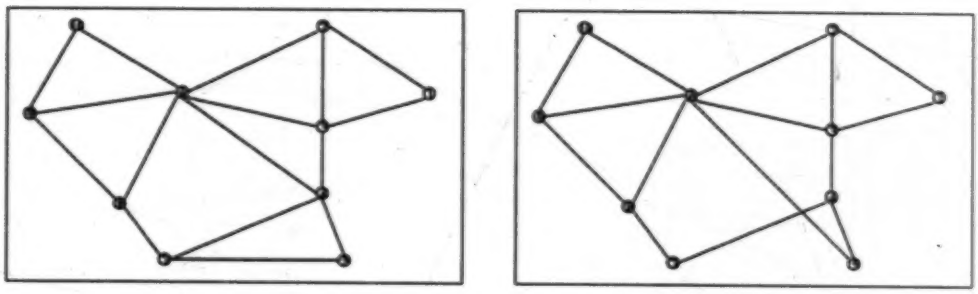


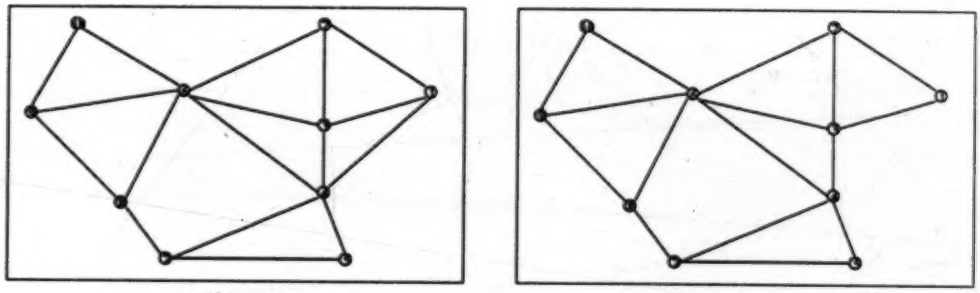
Figure 4.3: The Comparison of Link Percentage between the Two Algorithms



(*a*) (*b*)

Figure 4.4: The Layout of Physical Topology for EN-1 under Uniform Traffic with $\rho = 0$ and $\alpha = 0$: (*a*) Generated by Algorithm 1, (*b*) Generated by Algorithm 2

Figure 4.1 shows that the total cost of topology generated by Algorithm 1 is almost the same as that of Algorithm 2 for the same values of ρ and α. From Figure 4.2, we find that the running time of Algorithm 1 remains the same for different values of ρ and α, but that of Algorithm 2 depends heavily on α and a little on ρ. Except for the topologies with $\alpha = 0$, the running times spent Algorithm 1 are significantly less than those of Algorithm 2. In fact, when α becomes larger, the physical networks generated by the two algorithms will be denser. It can be seen in the model (3.2.1)-(3.2.4) that large values of α put more weight on the secondary routes in the minimization of cost function, resulting in a dense network to obtain short secondary routes. Since Algorithm 2 starts with a minimal 2-connected graph, it needs more iterations in the Link Addition Phase. However, Algorithm 1 uses the complete graph as the initial network, so that the Link Deletion Phase is the dominating phase. Thus, a denser final physical topology implies fewer iterations in the Link Deletion Phase if Algorithm 1 is used, but more iterations in the Link Addition Phase if Algorithm 2 is used. As shown in last chapter, each iteration of the Link Deletion Phase takes much less time than each iteration of the Link Addition Phase due to the speed up in computation. Consequently, the running time of the second algorithm increases



(a) (b)

Figure 4.5: The Layout of Physical Topology for EN-1 under Uniform Traffic with $\rho = 0.3$ and $\alpha = 0.1$: (*a*) Generated by Algorithm 1, (*b*) Generated by Algorithm 2

sharply and that of the first algorithm decreases slightly, when α becomes large.

Figure 4.3 indicates that link percentages of these topologies generated by Algorithm 1 are not much different from those by the second algorithm. This fact can be seen more clearly in the layouts of the generated network topologies in Figures 4.4-4.6.

From the above pictures, we can conclude that the two algorithms usually generate almost the same physical topology (at most differing in one or two links) for the same values of ρ and α. For most networks, the first algorithm spends less CPU time than the second one. The only exception is when $\alpha = 0$, that is, the case that no capacity is assigned to a secondary route for each source pair. We have also conducted computational experiments for other sample networks of up to 23 nodes and with non-uniform traffic. These experiments also show similar results and selected results are listed in Table 4.2. Therefore, we recommend the first algorithm for our design model if there are secondary routes ($\alpha > 0$).

To show the quality of solutions for the two algorithms, we use a branch and bound method to obtain the global optimal topology. Since the branch and bound method has exponential complexity and can only handle the network with up to

Table 4.2: Network Costs and Link Percentages of Selected Topologies Generated by the Two Algorithms for Other Sample Networks

Sample Network	ρ	α	Network Cost		Link Percentage	
			Algo.g 1	Algo. 2	Algo. 1	Algo. 2
Metro	0.0	0.0	2745.04	2745.04	0.285714	0.285714
	0.1	0.2	3958.87	3944.56	0.418532	0.392857
	0.2	0.3	4495.64	4546.03	0.474351	0.464286
	0.5	0.5	5764.28	5844.60	0.571429	0.571429
NSFNET	0.0	0.0	5567.07	5726.10	0.303030	0.303030
	0.1	0.2	7374.95	7374.95	0.333333	0.333333
	0.5	0.5	11126.58	11144.86	0.393939	0.386364
EN-2	0.0	0.0	9982.95	9974.20	0.235294	0.215686
	0.1	0.2	13143.35	13150.71	0.326797	0.294118
	0.1	0.3	14169.51	14193.78	0.333333	0.320261
	0.1	0.4	15181.08	15297.83	0.346405	0.349673
	0.1	0.5	16188.56	16294.33	0.366013	0.352941
	0.5	0.5	19669.14	19678.51	0.372549	0.359477
EN-3	0.1	0.1	18982.83	18980.37	0.233202	0.229249
	0.1	0.2	20710.14	20701.23	0.252964	0.249012
	0.1	0.3	22361.30	22355.81	0.274704	0.268775
	0.1	0.4	23986.89	23985.10	0.280632	0.280632
	0.1	0.5	25597.43	25587.96	0.288538	0.288538
	0.5	0.5	31170.94	31285.14	0.296443	0.300395

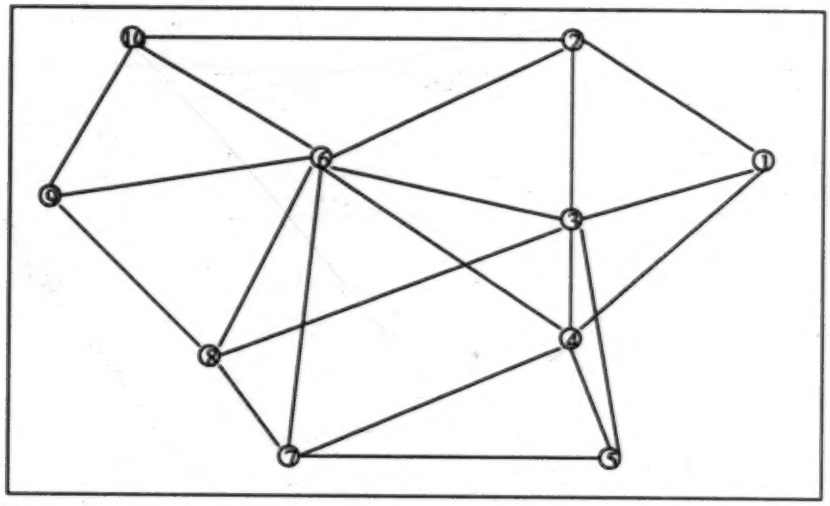


Figure 4.6: The Layout of Physical Topology for EN-1 under Uniform Traffic with $\rho = 0.5$ and $\alpha = 0.5$: Both Algorithms Produce the Same Topology

8 nodes in our computer facility, here we only use the sample network METRO. When $\rho = 0$ and $\alpha = 0$, the branch and bound method generates the global optimal network with cost of 2745.044128, which is the same as the network generated by Algorithm 1 and Algorithm 2. Changing ρ will not change the topology but just the link capacity. Therefore, for small networks, our two algorithms will produce very good local minimum.

4.2 Static Evaluation for the Topology Design Model

In this section, we only use Algorithm 1 to generate the ATM backbone topology since it takes less time than Algorithm 2 for most cases. The goal of our design model is to create the physical topology which helps the VP reconfiguration and dynamic bandwidth reallocation at the logical level. Statically, a physical network should provide short multiple routes and more extra capacity to support the adaptability to

the traffic fluctuation and facility failures in the logical VP level. Furthermore, the physical topology generated by the heuristic algorithm can be controlled by the two parameters, ρ and α, in the model (3.2.1)-(3.2.4). In the following, we evaluate the physical topology for these aspects and show how to adjust the two parameters in the design model to obtain certain required properties in the network topology.

The following measures of link capacity usage and topological properties are used to evaluate the physical topology.

- **Average link utilization**: The utilization of a link is defined as the ratio of actual bandwidth used by all the VPs passing through the link over the total bandwidth allocated on this link. The average link utilization gives a measure of bandwidth usage in the network under the current traffic requirements.

- **Maximum link utilization**: This measure gives the maximum possible bandwidth usage in the network. The difference between the **Maximum link utilization** and **Average link utilization** measures the variation of bandwidth usage in the network under the current traffic requirements.

- **Average hop count of primary route**: The first shortest path is used for the primary route. The average hop count should be small to reduce the VP set-up cost and the call routing cost.

- **Average hop count of secondary route**: The second shortest path is used for the secondary route if the primary route is overloaded. This should also be small.

- **Link percentage**: This is defined as the ratio of the number of established transmission links over the number of all possible links. This measures the sparseness of a network. For a large network, the link percentage should be small to obtain low link set-up cost. However this may cause large average hop counts.

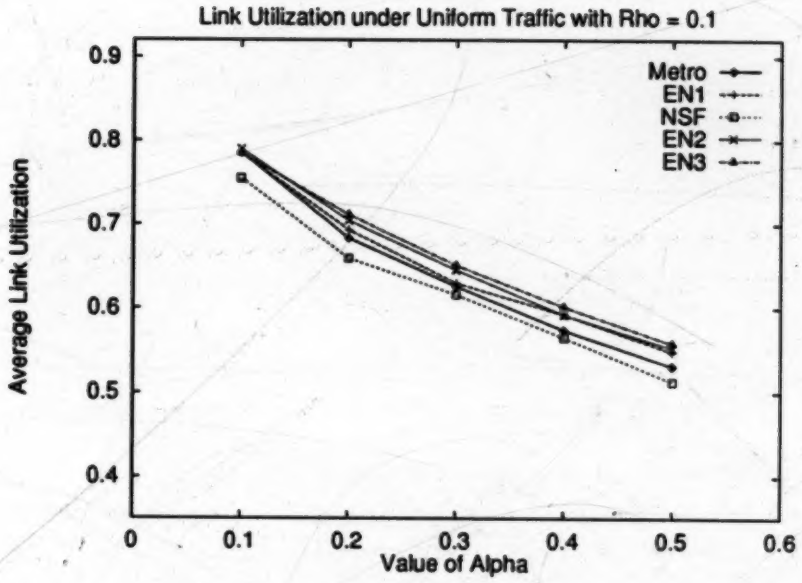


Figure 4.7: Average Link Utilization versus α under Uniform Traffic

The two parameters ρ and α in the design model (3.2.1)-(3.2.4) not only affect the total cost but also have impact on the determination of the two node-disjoint routes between each source pair. They can be tuned to adjust the above measures of the generated network for given locations of switch nodes and traffic requirements. In general, larger values of ρ and α imply that more spare capacity is assigned to the primary and secondary routes respectively, resulting in a generated network with higher cost. In the computational experiments, we will investigate the typical relationships between the two parameters and the above defined measures.

4.2.1 Static Evaluation under Uniform Traffic

The switch node locations from the five sample networks: METRO (8 nodes), NSF (12 nodes), EN-1 (10 nodes), EN-2 (18 nodes), and EN-3 (23 nodes), are considered for the heuristic algorithm with uniform traffic. In our experiments, the values of ρ

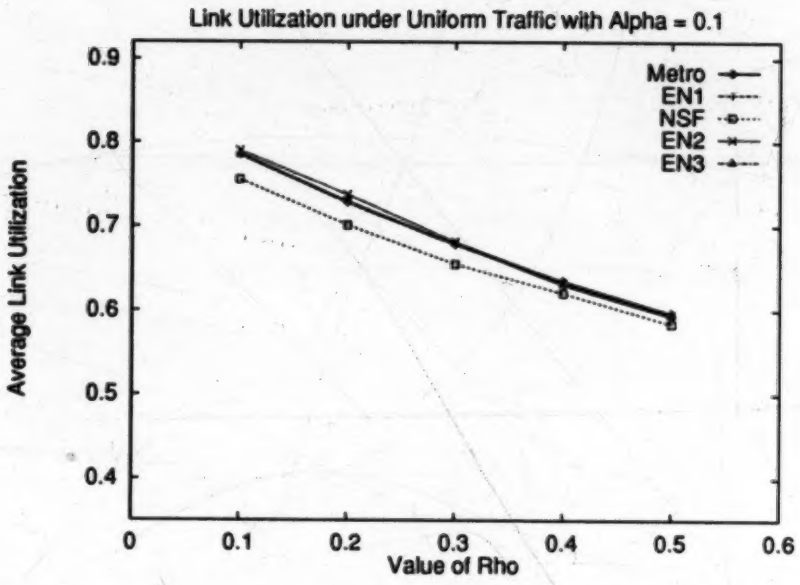


Figure 4.8: Average Link Utilization versus ρ under Uniform Traffic

and α are taken in the range [0, 0.5], since larger values cause too much capacity to be wasted.

The average link utilization is inversely related to the total cost, which can be easily seen from the design model (3.2.1). Figures 4.7 and 4.8 show that the values of ρ and α over [0, 0.5] produce reasonable link utilization. Although the average link utilization decreases with increasing ρ and α, it is important to note that the size of a network has little impact on the average link utilization. This means that the same values of ρ and α can be used in different networks to generate physical topologies with roughly the same level of link utilization.

Figure 4.9 shows that the maximum link utilization does not depend much on α and differs a little for different sample networks. It is worth noting that all the maximum link utilizations take values a little lower than 0.91. Since ρ specifies the extra capacity on the primary route, the maximum link utilization must be less than

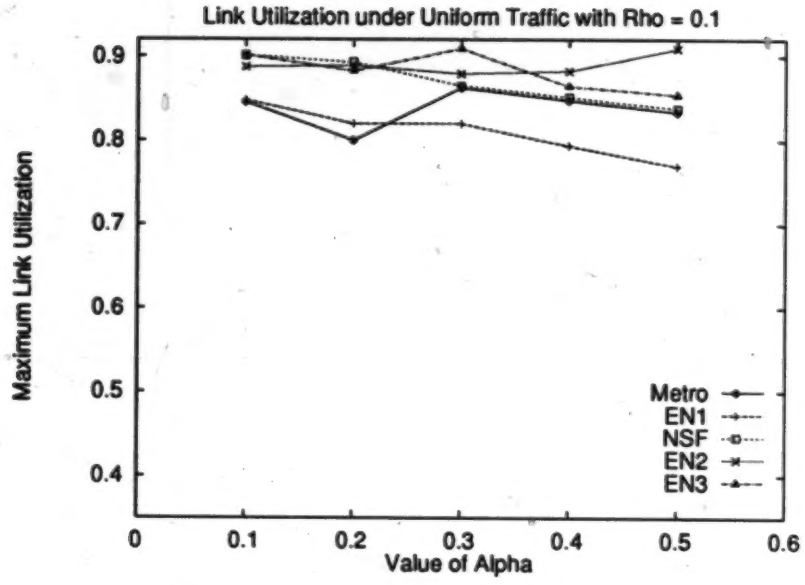


Figure 4.9: Maximum Link Utilization versus α under Uniform Traffic

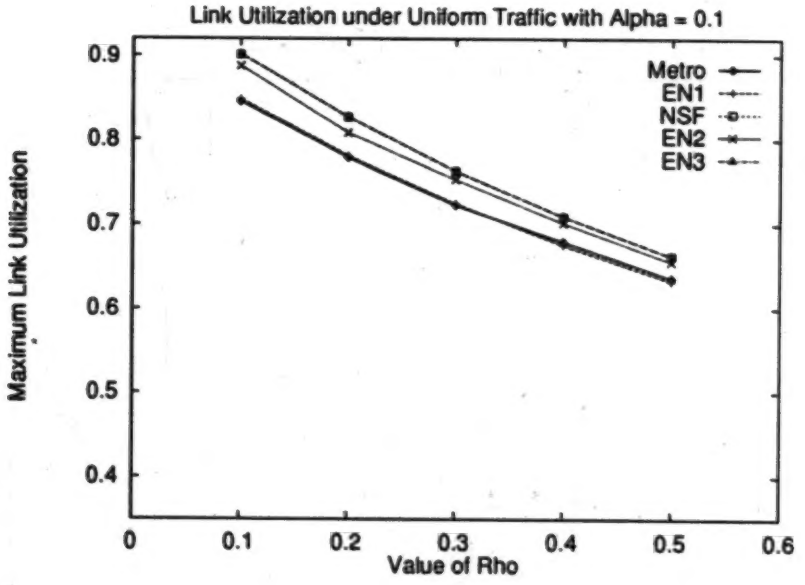


Figure 4.10: Maximum Link Utilization versus ρ under Uniform Traffic

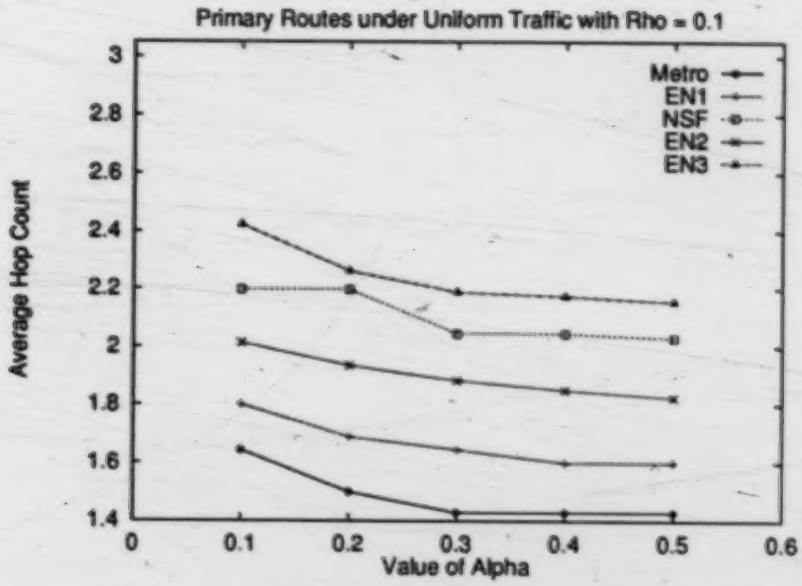


Figure 4.11: Average Hop Count of Primary Routes versus α under Uniform Traffic

$\frac{1}{1+\rho}$, which is 0.91 for $\rho = 0.1$. For large values of α, the maximum utilization is significantly higher than the average utilization and there exists some links which have no secondary routes passing through. However, Figure 4.10 shows that the maximum link utilization decreases monotonically with increasing ρ, which is to be expected.

From Figure 4.11, we can see that average hop count of the primary routes decreases with α. However, Figure 4.12 indicates that there is only a little variation with ρ and no strong trend up or down. Figures 4.11 and 4.12 also shows that average hop count of the primary routes typically increases with the size of a network. But there exists an exceptional case where the average hop counts of the primary routes for EN2 are smaller than those for NSF although EN2 has more nodes than NSF. This exception indicates that the hop count for a physical network not only depends on the size of the network, but also on other factors such as the distribution

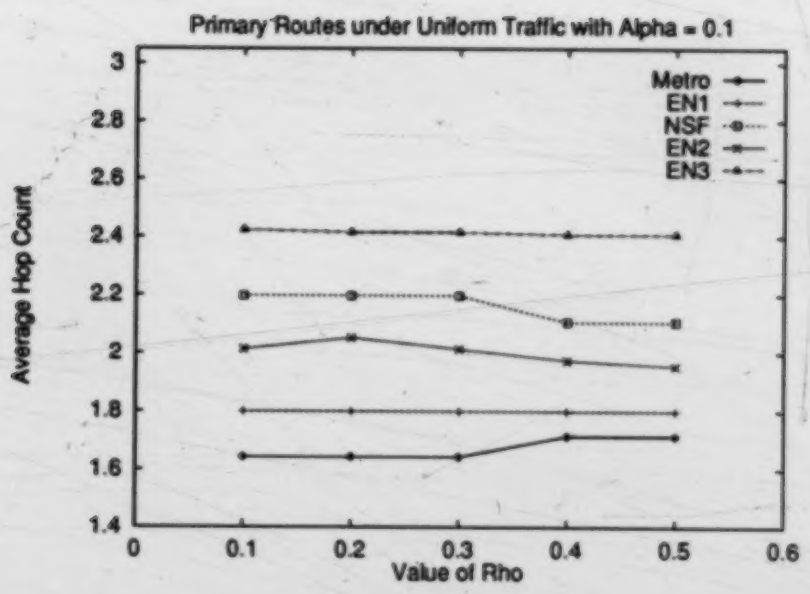


Figure 4.12: Average Hop Count of Primary Routes versus ρ under Uniform Traffic

of node locations. We should note that the hop count of the primary route drops relatively sharply on average when the value of α changes from 0.1 to 0.3. When α becomes larger than 0.3, however, little decrease in the hop count can be achieved. This implies that by setting $\alpha = 0.3$ better link utilization can be obtained with little increase in hop count for the primary route. In the call-level simulation study in the next section, we also find that large value of α does not reduce the call blocking rate of the dynamic capacity reallocation.

The results for average hop count of the secondary routes are similar to the primary route case as seen in Figures 4.13 and 4.14. Again, the average hop count is affected more by α than by ρ, but in this case it decreases with both α and ρ. The average hop count of secondary routes also varies more with α from 0.1 to 0.3 than from 0.3 to 0.5. As ρ is varied, this trend is less clear and it slightly decreases with ρ in some cases. The average hop count of secondary routes is longer than that of primary routes.

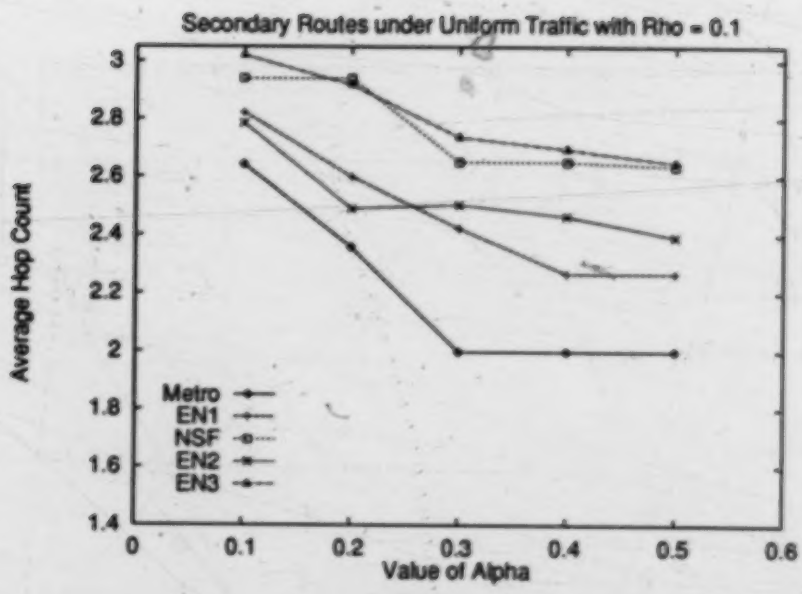


Figure 4.13: Average Hop Count of Secondary Routes versus α under Uniform Traffic

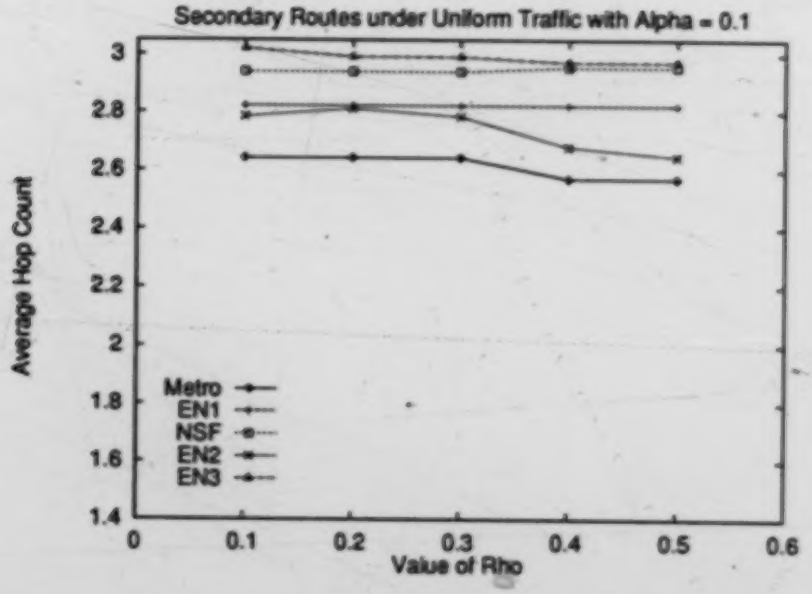


Figure 4.14: Average Hop Count of Secondary Routes versus ρ under Uniform Traffic

The link percentage of a network measures how dense the network is. Figures 4.15 and 4.16 show that the link percentage increases with α but typically is not dependent on ρ. This is because in the model (3.2.1)-(3.2.4), a large value of α puts more weight on the secondary routes in the minimization of the cost function, resulting in a denser network to obtain short secondary routes. We can also see that the link percentage becomes smaller as the size of a network gets larger, which is reasonable since we want a sparse topology for large networks anyway.

In general, varying α has a stronger effect on the performance measures than varying ρ. In the design model (3.2.1), ρ controls the spare capacity to the primary routes and α gives spare capacity to the alternate routes. Since the secondary routes are typically longer than the primary routes, increasing α by a unit will result in more bandwidth allocated in the links than increasing ρ by a unit. Thus, increasing α causes more capacity cost than increasing ρ and more changes in the topology can be achieved by varying α than ρ. The above computational results support this expectation.

4.2.2 Static Evaluation under Non-Uniform Traffic

The results for non-uniform traffic are similar to those for uniform traffic. Here we only list typical plots for link utilization, average hop counts of primary and secondary routes, and link percentage for the NSFNET sample network. Figure 4.17 shows that the average link utilization is monotonically decreasing with both α and ρ. However, it drops down sharply for small values of α and varies only slightly after $\alpha = 0.2$. This situation is different from that of the uniform traffic case where the average link utilization has a more linear behavior. Note that the traffic demands for the non-uniform case are normalized such that the total load is equal to that for the uniform traffic. Thus, the difference implies that the average link utilization is lower for non-uniform traffic in order to accommodate more variable traffic.

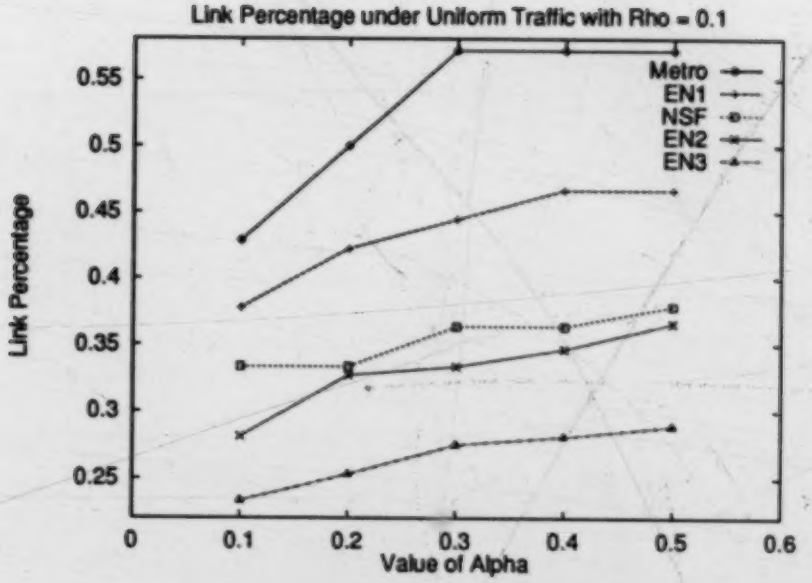


Figure 4.15: Link Percentage versus α under Uniform Traffic

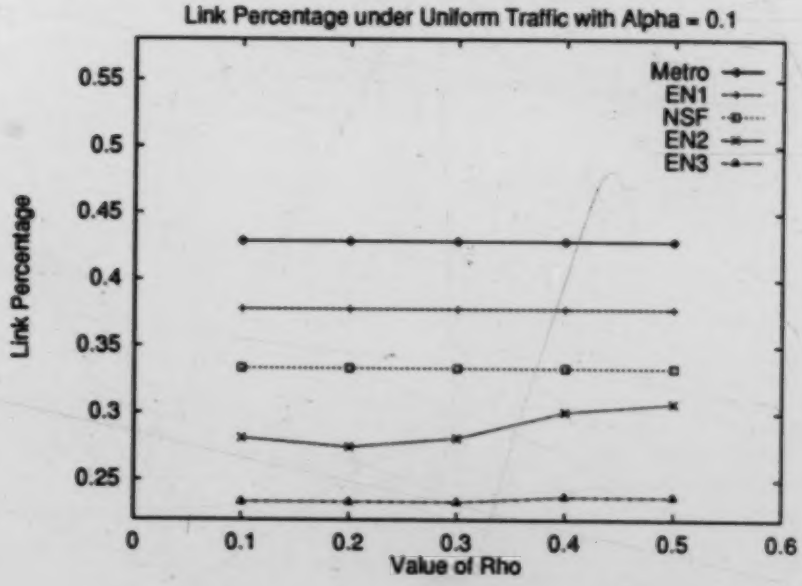


Figure 4.16: Link Percentage versus ρ under Uniform Traffic

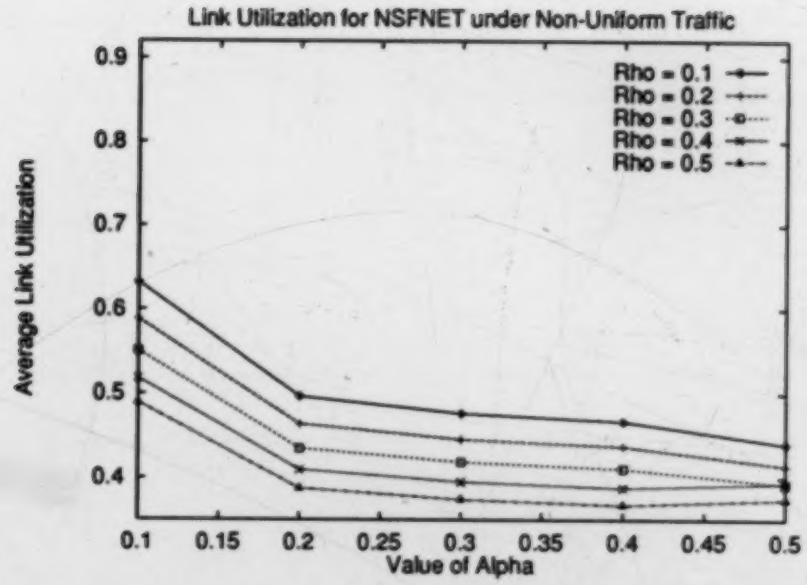


Figure 4.17: Average Link Utilization for NSFNET under Non-Uniform Traffic

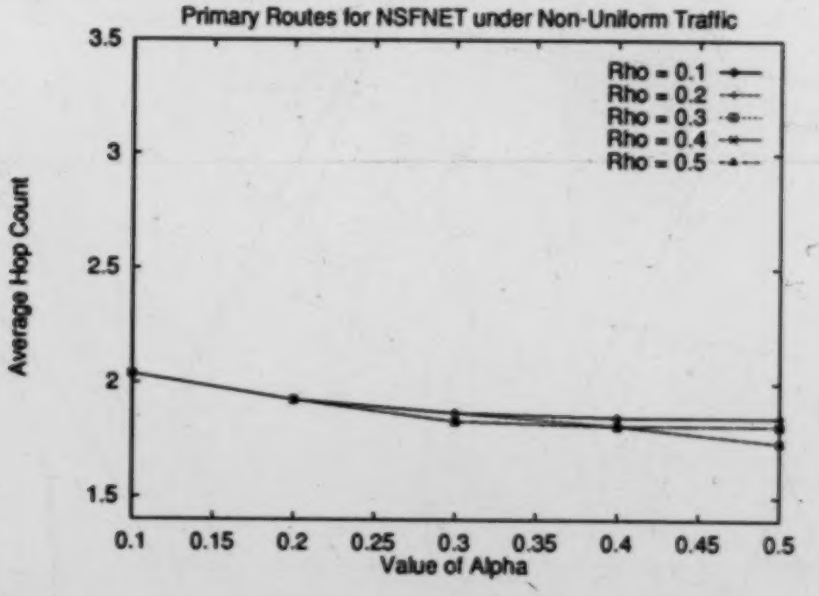


Figure 4.18: Average Hop of Primary Routes for NSFNET under Non-Uniform Traffic

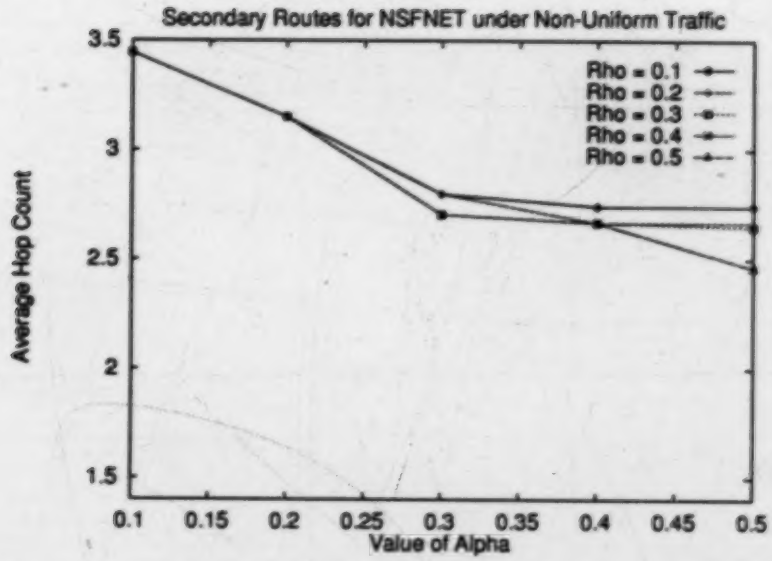


Figure 4.19: Average Hop of Secondary Routes for NSFNET under Non-Uniform Traffic

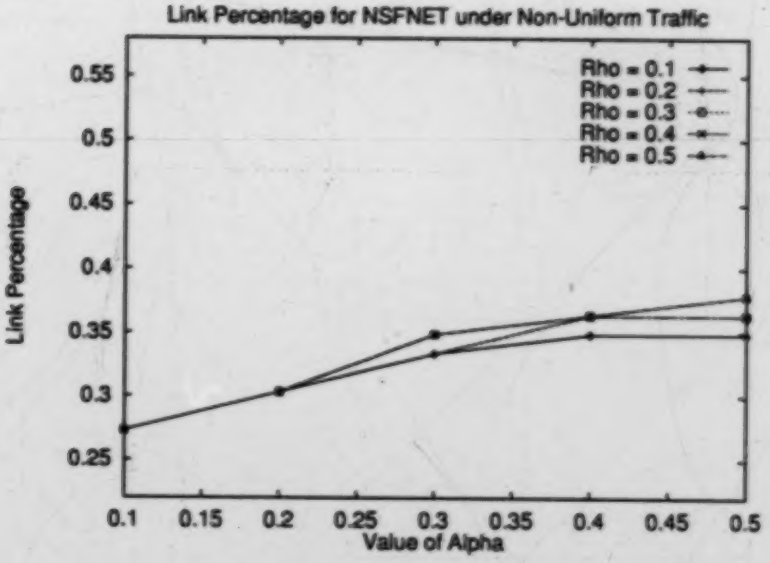


Figure 4.20: Link Percentage for NSFNET under Non-Uniform Traffic

Figures 4.18 and 4.19 show that the average hop count of primary and secondary routes decreases with increasing α but changes a little with ρ. Also Figure 4.20 indicates that link percentage of the network increases with α but it is affected only a little by varying ρ. Comparing these plots with those for NSF under uniform traffic, we find that these performance measures are not much different. Therefore, we conclude that the two parameters ρ and α affect the physical topology in the same way for both uniform and non-uniform traffic. The physical network generated by the heuristic algorithm is not much affected by the type of traffic.

4.3 Dynamic Evaluation for the Topology Design Model

The topology design model presented in the last chapter generates physical topologies for ATM backbone networks, which provides flexibility to the dynamic resource management at the logical VP level by using alternate paths and extra capacity. The logical VP management scheme adapts to changing traffic conditions and prevents possible network failures in a multimedia traffic environment. In this section we investigate the effect of the physical design parameters α and ρ on the performance of a logical VP network configuration using a call-level simulation. Based on the generated physical network, a logical VP network configuration can be constructed to accommodate the current traffic in the network. Two VP capacity allocation schemes will also be compared in the logical VP network configuration.

The call-level simulator is a C++ program developed at the University of Waterloo [41]. This simulator is used to test various routing and VPC management algorithms. To run the simulator, one should specify as the input the following six files: *Source*, *Config*, *Route*, *VPM*, *Seeds* and *Length*. The *Source* file contains the traffic characteristics for all source pairs. The *Config* file consists of the configurations

of both the physical network and the logical VP network. The *Route* file just contains a string which specifies the routing algorithm used in the simulation. The *VPM* file specifies the type of VPC management algorithm to be used. The *Seeds* file contains a batch of random seeds. The *Length* file specifies the total simulation length (in tick time) and the length of the transient period.

The first heuristic algorithm of Chapter 3 is used to find the capacity and placement of links (i.e., find the physical topology). A logical VP configuration is created by assigning a VP to each primary or secondary route. When a call arrives, if the primary route is not available (no capacity), the simulation will check the secondary route. If the secondary path is not available, the call is blocked. The **average call blocking rate**, defined as the fraction of blocked calls, is used as the performance measure to evaluate the logical VP configuration based on the generated physical network.

In our simulation, we carry out five runs to obtain an estimate of the **average call blocking rate** with a 95% confidence interval. The length of the transient period for each run is one fifth of the total simulation time. These set-ups produce reasonable accuracy for the estimate of the **average call blocking rate**, as shown later on.

The traffic requirements for the physical network design are converted into the traffic parameters for this call-level simulation. The traffic characteristics for each source pair are described by the *mean inter-arrival time*, the *mean duration time* and the *call peak rate*. We assume that the call inter-arrival time and call service time both follow a geometric distribution and that the call peak rate is deterministic.

One question of interest is how well the physical design facilitates a bandwidth allocation scheme. For this simulation, we consider the **static** bandwidth allocation scheme and a **dynamic** bandwidth allocation scheme based on Baba's method [2]. In the first scheme, the primary VP for the source pair (k, h) is given capacity $(1+\rho)r_{kh}$ and the secondary VP is given capacity αr_{kh} using up all the capacity allocated

in the system. This capacity will not change throughout the simulation. For the dynamic capacity reallocation scheme, the primary VP of the source pair (k,h) has initial capacity r_{kh} and the secondary VP is assigned the initial capacity $0.1r_{kh}$. The remaining bandwidth in the network is shared by all VPs to adjust the capacity of each VP during the simulation. Baba's dynamic scheme requires the user to specify the *threshold* value for the spare bandwidth and the *step-size* for the bandwidth change in each VP. If the remaining unused capacity of a VP is less than the prespecified *threshold* value, then the VP is allocated *step-size* bandwidth. If a VP has more spare capacity than the *threshold* value, its capacity is reduced by *step-size*. The capacity of a VP can never be less than the initial capacity assigned. When the capacity of a VP is adjusted, the VP is locked for a prespecified amount (*delay*) of tick time and cannot be used by any call during that period. This is to simulate the processing time needed by the dynamic bandwidth allocation scheme.

4.3.1 Dynamic Evaluation under Uniform Traffic

For uniform traffic, we only present the simulation results for NSFNET sample network since experiments on other sample networks give similar results. We assume that all the 132 source pairs have the same amount traffic: $r_{kh} = 100Mbps$ for $k,h = 1,\cdots,12$. For each physical network generated by the first heuristic algorithm, the static bandwidth allocation scheme assigns the capacity $(1+\rho)100Mbps$ to each of the primary VPs and the capacity $\alpha 100Mbps$ to each of the secondary VPs. For each pair, the traffic characteristics are: *mean inter-arrival time* = 13 ticks, *mean duration time* = 100 ticks and *call peak rate* = $10Mbps$. This is dimensioned to keep the call blocking rate within 3% for moderate load in the network. The capacity of each physical link is determined by the heuristic algorithm of Chapter 3.

The simulation parameters for the dynamic bandwidth allocation scheme under uniform traffic are set as follows: For each source pair (k,h), the primary VP is as-

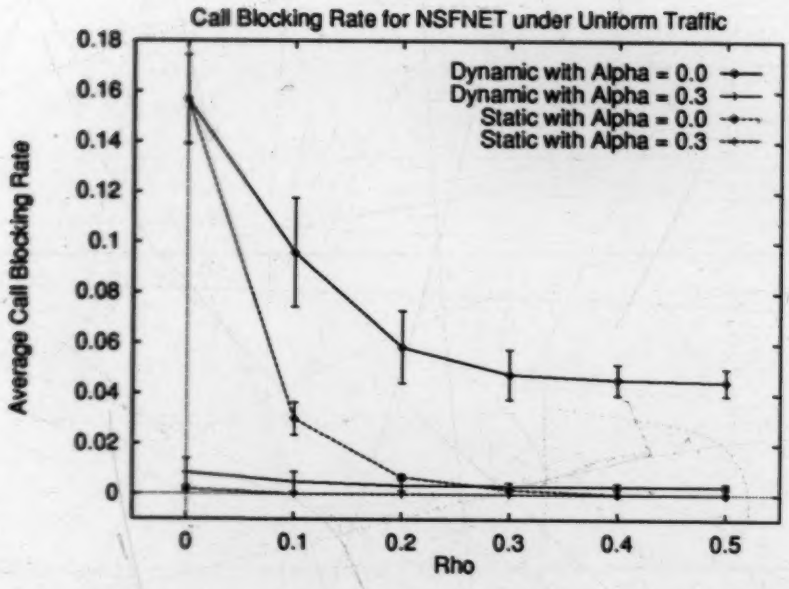


Figure 4.21: Average Call Blocking Rate versus ρ for NSFNET under Uniform Traffic

signed *initial bandwidth* = $r_{sh} = 100Mbps$ and the secondary VP has *initial bandwidth* = $10Mbps$; *step-size* = $15Mbps$, *threshold* = $10Mbps$ and *delay* = 1 tick.

In Figure 4.21, we can see that if there is no alternate route (i.e., $\alpha = 0$) the average call blocking rate drops significantly when more extra capacity is assigned to the primary route of each source pair (that is, increasing values of ρ). When an alternate route with a fixed capacity (i.e., $\alpha = 0.3$) is used for each source pair, the average call blocking rate is not reduced significantly by increasing the extra capacity on the primary routes. As shown in Sections 4.1 and 4.2, the physical topology changes very little when varying ρ with fixed $\alpha > 0$. Thus, increasing ρ only causes more extra capacity to be assigned to the primary routes. But when $\alpha = 0.3$, there is enough bandwidth in the alternate route for each source pair to reduce the average call blocking rate to a very low value (i.e., the current traffic is handled very well). In this situation, more extra capacity in the primary routes would not reduce the call

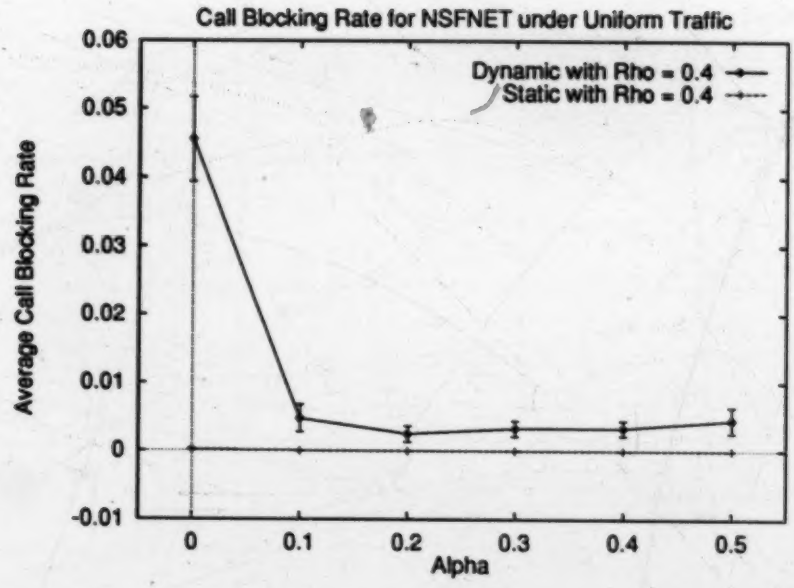


Figure 4.22: Average Call Blocking Rate versus α for NSFNET under Uniform Traffic

blocking rate further, resulting in a waste of network bandwidth.

From Figures 4.21 and 4.22 we can see that the static bandwidth allocation scheme always has smaller average call blocking rate than the dynamic bandwidth allocation scheme. For uniform traffic, each call has the same traffic characteristics. For the static scheme, the VPs are assigned the maximum available bandwidth. But the primary VP is assigned the *initial bandwidth = 100 Mbps* and the secondary VP the *initial bandwidth = 10 Mbps* for each source pair in the dynamic scheme, no matter how much extra capacity exists. Then the dynamic scheme will allocate the extra capacity to a VP when the remaining capacity is less than the *threshold = 10 Mbps*. Since the *threshold* can only accommodate one call and the *step-size* bandwidth is allocated after one *delay* time, the VP may result in some calls blocked if more than one calls arrive within the *delay*. Therefore, in general the static scheme achieves lower call blocking rate than the dynamic scheme.

The effect of varying α is shown in Figure 4.22 with fixed $\rho = 0.4$. The average call blocking rate for $\alpha = 0$ is much higher compared to those for $\alpha = 0.1$ to 0.4 in the dynamic scheme. This implies that only a small amount of capacity is needed for the secondary route for the dynamic allocation scheme to reduce the average call blocking rate in the network. The resulting reduction of the average call blocking rate comes from the change of the physical topology since increasing α from 0 to 0.1 changes the topology significantly, as shown in Sections 4.1 and 4.2. Without the alternate route, the dynamic scheme cannot handle the traffic well. This is evidence to support the design model in Chapter 3 that two routes should be provided to each source pair. Again the static bandwidth allocation scheme has better performance than the dynamic scheme for uniform traffic because no different bandwidth demands are needed to be balanced by the extra capacity in the network.

4.3.2 Dynamic Evaluation under Non-Uniform Traffic

For non-uniform traffic, the static allocation scheme cannot balance different bandwidth demands of source pairs so it causes higher call blocking rate than the dynamic bandwidth allocation scheme. We expect that the extra capacity in the network can be better utilized by the dynamic bandwidth allocation scheme due to the diversity of traffic demands. The simulation parameters for the dynamic scheme are set as follows:

- For each source pair (k, h), the primary VP is assigned *initial bandwidth* = r_{kh} and the secondary VP has *initial bandwidth* = $0.1r_{kh}$; *step-size* = $0.15r_{kh}$, *threshold* = $0.1r_{kh}$ and *delay* = 1 tick, where r_{kh} randomly takes from 2 to 1300 Mbps.

- There are 108 source pairs. That is, $|\{r_{kh} > 0\}| = 108$. For each pair, *mean inter-arrival time* = 20 ticks, *mean duration time* = 100 ticks and *call peak rate*

$= 0.1 r_{kh}$. The capacity of each physical link is determined by the first heuristic algorithm of Chapter 3.

In the first experiment, we investigate the effect of the two parameters ρ and α on the ability of the dynamic capacity reallocation scheme to handle the traffic in the network. Figures 4.23 and 4.25 display the performance of the static and dynamic allocation schemes for topologies generated by different values of α and ρ. The plots are obtained from multiple runs showing 95% confidence intervals. It is easy to see that the dynamic allocation scheme performs much better than the static scheme. This is because the dynamic scheme results in better utilization of the extra capacity due to the diversity of traffic demands under non-uniform traffic. The static allocation scheme assigns all available bandwidth to the VPs so that no extra capacity exists to handle the different bandwidth demands of the source pairs for the non-uniform traffic.

Figure 4.23 shows that for the dynamic allocation scheme, the average call blocking rate decreases when α varies from 0.0 to 0.3 and increases again when α becomes larger than 0.3. This indicates that adding too much capacity in the secondary VPs does not help the dynamic allocation scheme. This is because calls of different sizes with respect to bandwidth exist in the network. For small α, many calls of small bandwidth can be accommodated and calls of large bandwidth are rejected. When α increases a little, more small calls get accepted but large calls are still rejected. This results in the average call blocking rate being reduced. When α becomes much larger, more big calls get accepted resulting in the rejection of many small calls. Overall, the average call blocking rate increases again. This can be seen in the detailed simulation outputs where the utilizations for those VPs used by small calls become smaller and then get larger again when α keeps increasing. Figure 4.24 shows the average VP utilization versus α.

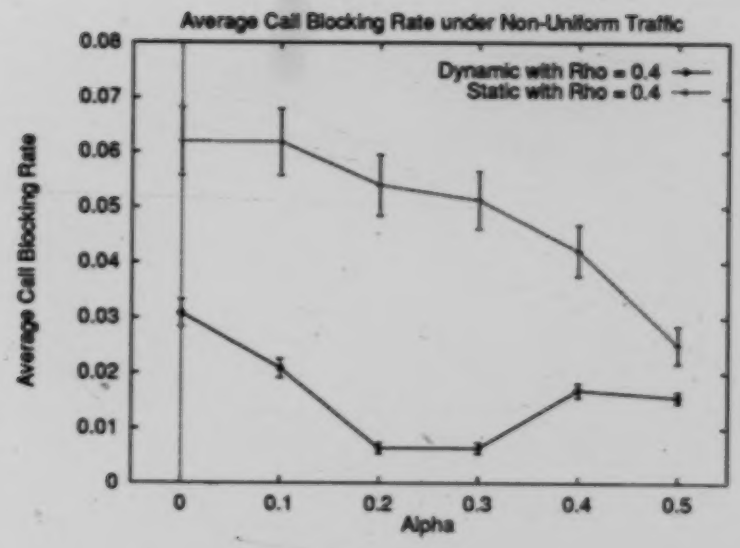


Figure 4.23: Average Call Blocking Rate versus α for NSFNET under Non-Uniform Traffic

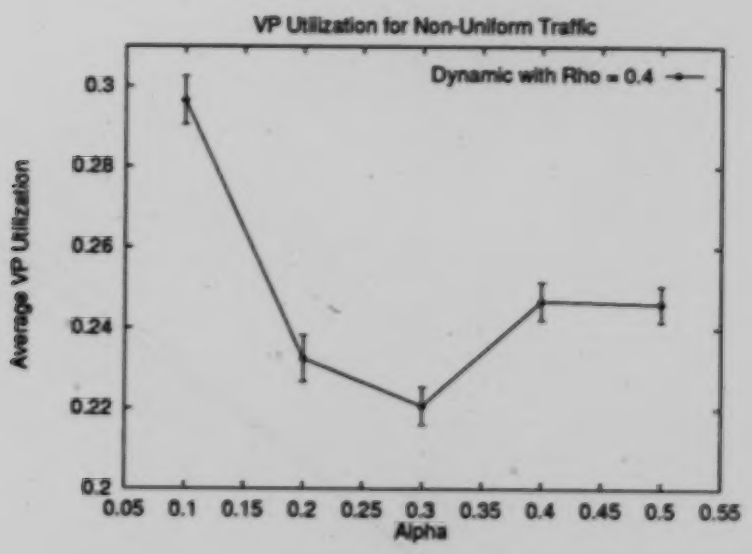


Figure 4.24: Average VP Utilization versus α for NSFNET under Non-Uniform Traffic

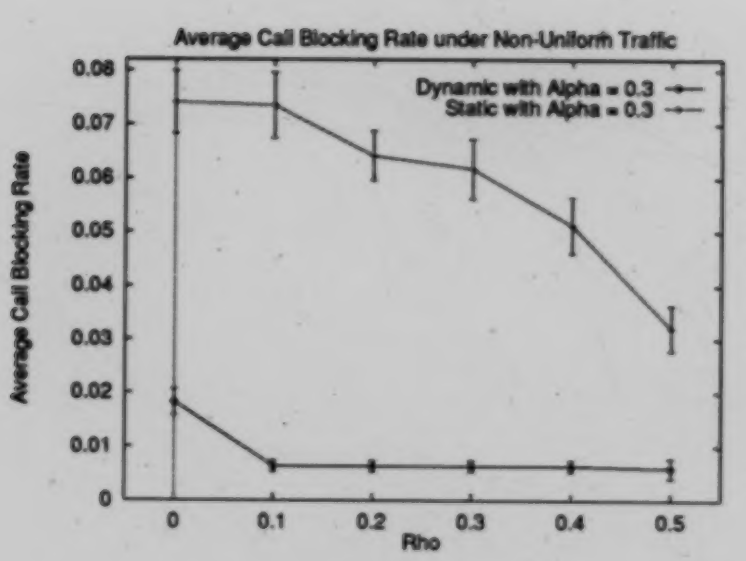


Figure 4.25: Average Call Blocking Rate versus ρ for NSFNET under Non-Uniform Traffic

In Figure 4.25, the average blocking rate for the dynamic scheme drops when ρ goes from 0.0 to 0.1 and it then becomes flat when ρ is larger than 0.1. This implies that only $0.1r_{sA}$ extra capacity is needed for the primary VP of each pair and more extra capacity cannot help the dynamic scheme any more. As the two parameters ρ and α become larger, the cost of the network increases. From this experiment, we conclude that the physical topology generated by our algorithm supports the dynamic bandwidth allocation scheme at only a small additional cost. The dynamic bandwidth allocation scheme is more appropriate for the network to handle the non-uniform traffic.

In the next experiment, we look at the ability of the physical design to help the dynamic reallocation scheme to handle traffic changes. Figure 4.26 gives a comparison between the static and the dynamic reallocation schemes for a particular physical topology with $\rho = 0.4$ and $\alpha = 0.3$ when the traffic load in the network is increased.

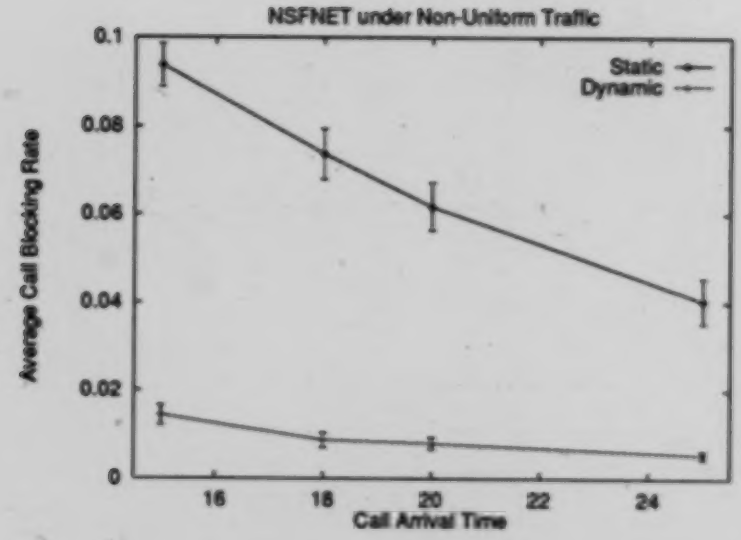


Figure 4.26: Average Call Blocking Rate versus the Traffic Load under Non-Uniform Traffic

This plot clearly shows that the dynamic scheme is resilient against changing traffic conditions compared to the static one. Many plots for other generated physical topologies also show the same result, although we do not list them here. This result suggests that the physical topology generated by our algorithm together with the dynamic bandwidth allocation scheme improve network efficiency.

Chapter 5

Conclusion and Future Work

5.1 Summary and Conclusion

The logical VP-based network provides an effective way to facilitate the dynamic resource management in ATM networks. The physical topology of an ATM network should be designed to provide flexibility to the logical VP reconfiguration in order to adapt to changing traffic conditions and possible network facility failures. In this thesis, we have considered the problem of designing a physical topology for ATM networks with a view to facilitating the dynamic resource management at the logical VP reconfiguration phase.

Chapter 1 presented the motivation for designing such a physical topology in ATM networks and introduced the design problem. In Chapter 2, we reviewed the main important techniques for physical topology design and for logical VP configuration design in general communication networks in the recent literature.

In Chapter 3, we addressed the problem of designing a physical topology for ATM networks in order to improve the efficiency of the dynamic resource management at logical VP level. The design model is formulated as a nonlinear mixed programming problem where the objective cost function to be minimized is defined by considering the balance between the link capacity cost and the link installation cost. Two

parameters ρ and α are introduced in the design model to control the capacity assignments and topological properties. There are two important properties for our design model. The first is that given the placement of transmission links in the network, the optimal routing is determined by the first two node-disjoint paths between each source pair. The second property is that the feasible physical topology must be at least 2-connected. These two properties have been exploited to develop two effective heuristic algorithms to solve the problem. The first heuristic algorithm starts with the complete graph, iterates first by link deletion and then by link addition while a cost improvement can be made. At each iteration, the 2-connectivity is preserved. The second algorithm starts with a minimal 2-connected graph, iterates first by link addition and then by link deletion to reduce the network cost. In the last two sections of this chapter, we developed an efficient method to check the 2-connectivity of a graph with removal of a link. Murchland's method [21] is used to modify the first shortest path between all source pairs after a link is deleted. We also extended Murchland's method to speed up the recalculation of the second node-disjoint shortest path between all source pairs after a link is deleted.

The two parameters, ρ and α, in the design model play an important role in the generated physical topology. The extra capacity on the generated network is controlled by ρ and α. In particular, ρ controls the amount of spare capacity to the primary routes and α gives spare capacity to the secondary routes. Therefore, they can be used in turn to tune the final topology for better network efficiency supplied by the dynamic resource management.

In Chapter 4, the performance of generated topologies was studied. We first made a comparison between our two heuristic algorithms through computational experiment on a typical sample network. The computational results show that the two algorithms usually generate almost the same physical topology (at most differing in one or two links) for the same values of ρ and α. Since each iteration for link deletion

takes less time than that of a link addition and the first algorithm mostly uses a link deletion while the second algorithm is dominated by link additions, usually the first algorithm spends less CPU time than the second one. The only exception is when $\alpha = 0$, which is the case that no capacity is assigned to secondary route for each source pair.

In the second part of Chapter 4, we evaluated the physical networks generated by the first heuristic algorithm in terms of link capacity usage, average hop counts for the two node-disjoint routes and link percentage. These performance measures statically determine the characteristics of the physical topology. The computational experiments based on five sample networks have shown that the two optimization parameters ρ and α can be adjusted properly in order to make a trade-off between the network cost and the network performance. The static evaluation has concluded that the second parameter α, which provides extra capacity to the alternative route, has greater influence on the physical topology than the first parameter ρ. In general, increasing ρ and α results in more extra capacity to be allocated in the network and in turn the physical network incurs more cost. In our design model, the physical network performance can be much improved by slightly increasing ρ and α while bringing in little increase of the network cost. In addition, the computational results showed that the physical topology designed using our model is not much sensitive to the type of traffic.

Finally, using a call-level simulation, we investigated the effect of the two design parameters on the ability of the dynamic bandwidth management in the logical VP network to handle network traffic. The average call blocking rate in the VP configuration, created by assigning a VP to each primary or secondary route in the physical network, is used to measure such an ability. The dynamic performance evaluation from the call-level simulation experiments reveals that the generated physical network with small extra capacity allocated (i.e., with small cost) can achieve the same

performance as that with large extra capacity, as shown in the static evaluation. The alternate route with a small to moderate amount of capacity is needed for the dynamic bandwidth allocation scheme to reduce the average call blocking rate in the network. The simulation results also show that the physical network designed by our model helps the dynamic bandwidth allocation scheme to handle the traffic load changes under the non-uniform traffic scenario.

In summary, the physical network designed by our model facilitates the dynamic resource management at the VP level in the following two ways:

- The physical topology is 2-connected and provides two node-disjoint routes to each source pair as well as some extra capacity on these routes. Thus the network has potential ability to adapt to the changing traffic conditions and to prevent possible network failures.

- The design model (3.2.1)-(3.2.4) includes two parameters (ρ and α) which can be tuned up to generate a physical topology with good performance and low cost.

5.2 Future Work

In the design model proposed in this thesis, traffic requirements of all source pairs are assumed to be given or to be obtained by some traffic models. In computational experiments we only considered a single class of traffic. For ATM networks it is very important to investigate how to obtain these traffic requirements for multiple classes of services and how well our design model handles multiple classes of traffic.

In this thesis, our design model is for backbone networks. We assume that the local access networks have been designed and regard a local access network as a node in the backbone network. Many methods have been proposed for the centralized access network attached to each backbone node [21]. It is interesting to investigate the

effect of a design method for local access network on the physical backbone topology generated by our model for the backbone network and its dynamic performance.

In Chapter 4, we only consider two VP bandwidth allocation schemes to dynamically evaluate the flexibility of the generated physical topologies. For uniform traffic, we found that the dynamic reallocation scheme worked poorly. It could be useful to look at other dynamic capacity reallocation schemes. When the traffic fluctuation is severe, more complicated VP reconfiguration algorithms (such as VP topology reconfiguration) may improve the network efficiency much more than the VP bandwidth reallocation schemes, but maybe more disruptive to the operation of the network. It is worth studying the effect of our design model on the performance of the VP reconfiguration algorithms.

The complexity of the two heuristic algorithms in this thesis can be reduced by employing advanced data structures. For example, the adjacency list and heap sorting of shortest neighbors can be used to reduce the complexity of Dijkstra's algorithm. Furthermore, more efficient methods to modify the second shortest path between all source pairs, especially in the case of link addition, is desired to speed up these two algorithms.

Bibliography

[1] S. Ahn, R. P. Tsang, S. R. Tong and D. H. C. Du. Virtual Path Layout Design on ATM Networks, in *Proc. IEEE INFOCOM*, pp. 192-200, 1994.

[2] K. I. Baba, M. Murate, and H. Miyahara. Performance Analysis for the Dynamic Virtual Path Bandwidth Control Method in ATM Networks, *International Journal of Communication Systems*, vol. 7, no. 4, pp. 283-94, 1994.

[3] D. Bertsekas and R. Gallager. *Data Networks, 2nd ed.*, Englewood Cliffs, NJ: Prentice-Hall, 1992.

[4] Nanda G. Chattopadyay, Thomas W. Morgan, and A. Raghuram. An Innovative Technique for Backbone Network Design, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1122-32, Sept./Oct. 1989.

[5] Sung-hark Chung, Young-soo Myung and Dong-wan Tcha. Optimal Design of a Distributed Network with a Two-Level Hierarchical Structure, *European Journal of Operational Research*, vol. 62, pp. 105-15, 1992.

[6] Martin De Prycker. *Asynchronous Transfer Mode*, 2nd. ed., New York: Ellis Horwood, 1993.

[7] Reuven Elbaum and Sidi Moshe. Topological Design of Local Area Networks Using Genetic Algorithms, in *Proc. IEEE INFOCOM*, pp. 64-71, 1995.

[8] Cem Ersoy and Panwar Shivendra. Topological Design of Interconnected LAN-MAN networks, *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 8, pp. 1172-82, Oct. 1993.

[9] A. Farago, S. Blaabjerg, L. Ast, G. Gordos and T. Henk. A New Degree of Freedom in ATM Network Dimensioning: Optimizing the Logical Configuration, *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1199-1205, Sept. 1995.

[10] V. J. Friesen, J. J. Harms, and J. W. Wong. Resource Management with Virtual Paths in ATM Networks, *IEEE Network*, pp. 10-20, Sept. 1996.

[11] B. Gavish. Topological Design of Computer Communication Networks - The Overall Design Problem, *European Journal of Operational Research*, vol. 58, pp. 149-72, 1992.

[12] Mario Gerla, Jose A. Monteiro, and Rodelfo Pazos. Topology Design and Bandwidth Allocation in ATM Nets, *IEEE Journal on Selected Areas in Communications*, vol. 7, pp. 1253-62, 1989.

[13] Alexander Gersht and Robert Weihmayer. Joint Optimization of Data Network Design and Facility Selection, *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 1667-81, 1990.

[14] O. Gerstel and A. Segall. Dynamic Maintenance of the Virtual Path Layout, in *Proc. IEEE INFOCOM*, pp. 330-337, 1995.

[15] W. D. Grover. The Self-Healing Network: A Fast Distributed Restoration Technique for Networks Using Digital Cross-Connect Machines, in *Proc. IEEE GLOBECOM*, pp. 1090-95, Dec. 1987.

[16] Roch Guerin, Hamid Ahmadi and Mahmoud Naghshineh. Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks, *IEEE J. Select. Areas Commun.*, vol. 9, pp. 968-81, 1991.

[17] Alan Herschtal and Meir Herzberg. Dynamic Capacity Allocation and Optimal Rearrangement for Reconfigurable Networks, in *Proc. IEEE GLOBECOM*, pp. 946-51, 1995.

[18] W. Hsieh, M. Gerla, P. McGregor, and J. Eckl. Locating Backbone Switches in a Large Packet Network, in *Proceedings of the 3rd International Conference on Computer Communication*, Toronto, pp. 330-39, 1976.

[19] J. Y. Hui, M. B. Gursoy, N. Moayeri, and R. D. Yates. A Layered Broadband Switching Architecture with Physical or Virtual Path Configurations, *IEEE J. Select. Areas Commun.*, vol. 9, pp. 1416-26, 1991.

[20] A. Iwata, R. Izmailov, D. S. Lee, B. Sengupta, G. Ramamurthy and H. Susuki. ATM Routing Algorithms with Multiple QoS Requirements for Multimedia Internetworking, *IEICE Trans. Commun.*, vol. E79-B, no. 8, pp. 999-1007, Aug. 1996.

[21] Aaron Kershenbaum. *Telecommunications Network Design Algorithms*, McGraw-Hill, 1993.

[22] Aaron Kershenbaum, Parviz Kermani, and George A. Grover. Mentor: An Algorithm for Mesh Network Topological Optimization and Routing. *IEEE Transactions on Communications*, vol. 39, pp. 503-13, 1991.

[23] Jae-gyun Kim and Dong-wan Tcha. Optimal Design of a Two-Level Hierarchical Network with Tree-Star Configuration, *Computers and Industry Engineering*, vol. 22, no. 3, pp.273-81, 1992.

[24] S. B. Kim. An Optimal Establishment of Virtual Path Connections for ATM Networks, in *Proc. IEEE INFOCOM*, pp. 72-79, 1995.

[25] Chang-ho Lee, Hyung-bong Ro and Dong-wan Tcha. Topological Design of a Two-Level Network with Ring-Star Configuration. *Computers and Operations Research*, vol. 20, pp. 625-37, 1993.

[26] Ming-jing Lee and James R. Yee. Optimal Minimax Routing in ATM Networks, in *Proc. IEEE GLOBECOM*, pp. 869-73, 1990.

[27] Ming-jing Lee and James R. Yee. A Partial Branch and Bound Design Algorithm for Reconfigurable Networks, in *Proc. IEEE ICC*, pp. 682-6, 1991.

[28] Ming-jing Lee and James R. Yee. A Design Algorithm for Reconfigurable ATM networks, in *Proc. IEEE INFOCOM*, pp. 144-51, 1993.

[29] Ming-jing Lee and James R. Yee. A Topology and Discrete Capacity Assignment Algorithm for Reconfigurable Networks, *Operations Research*, vol. 43, pp. 102-16, 1995.

[30] F. Y. S. Lin and K. T. Cheng. Virtual Path Assignment and Virtual Circuit Routing in ATM Networks, in *Proc. IEEE GLOBECOM*, pp. 436-41, 1993.

[31] Michael Logothetis and Shigeo Shioda. Centralized Virtual Path Bandwidth Allocation Scheme for ATM networks, *IEICE Trans. Commun.*, vol. E75-B, no. 10, pp. 1071-80, Oct. 1992.

[32] D. Medhi. Multi-Hour, Multi-Traffic Class Network Design for Virtual Path-Based Wide-Area Dynamically Reconfigurable ATM Networks, *IEEE/ACM Transactions on Networking*, vol. 3, no. 6, pp. 809-18, Dec. 1995.

[33] D. Medhi and C. T. Lu. Dimensioning and Computational Results for Wide-Area Broadband Networks with Two-Level Dynamic Routing, *IEICE Trans. Commun.*, vol. E80-B, no. 2, pp. 273-281, Feb. 1997.

[34] J. A. Monteiro and Mario Gerla. Topological Reconfiguration of ATM Networks, in *Proc. IEEE INFOCOM*, pp. 207-214, 1990.

[35] K. Murakami and H. S. Kim. Virtual Path Routing for Survivable ATM Networks, *IEEE/ACM Transactions on Networking*, vol. 4, no. 1, pp. 22-39, Feb. 1996.

[36] H. Nakamura and T. Oda. Optimization of Facility Planning and Circuit Routing for Survivable Transport Networks - An Approach Based on Genetic Algorithm and Incremental Assignment, *IEICE Trans. Commun.*, vol. E80-B, no. 2, pp. 240-251, Feb. 1997.

[37] S. Ohta and K. Sato. Dynamic Bandwidth Control of Virtual Path in an Asynchronous Transfer Mode Network, *IEEE Trans. Commun.*, vol. 40, pp. 1239-47, 1992.

[38] E. Oki and N. Yamanaka. An Optimum Logical-Design Scheme for Flexible Multi-QoS ATM Networks Guaranteeing Reliability, *IEICE Trans. Commun.*, vol. E78-B, no. 8, pp. 1016-24, July 1995.

[39] R. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues*, Archtech House, Inc., 1994.

[40] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Inc., 1982.

[41] R. Quershi and J. Harms. A Flexible Virtual Path Topology Design Algorithm, *accepted by Proc. IEEE GLOBECOM*, 1997.

[42] B. H. Ryu, H. Ohsaki, M. Murata, and H. Miyahara. Design Algorithm for Virtual Path Based ATM Networks, *IEICE Trans. Commun.*, vol. E79-B, no. 2, pp. 97-107, Feb. 1996.

[43] K. I. Sato, S. Ohta and I. Tokizawa. Broad-band ATM Network Architecture Based on Virtual Paths, *IEEE Trans. Commun.*, vol. 38, pp. 1212-22, 1990.

[44] S. Shioda and H. Uose. Virtual Path Bandwidth Control Method for ATM Networks: Successive Modification Method, *IEICE Trans.*, vol. E74, no. 12, pp. 4061-68, Dec. 1991.

[45] S. Shioda, H. Saito, and H. Yokoi. Sizing and Provisioning for Physical and Virtual Path Networks Using Self-Sizing Capability, *IEICE Trans. Commun.*, vol. E80-B, no. 2, pp. 252-262, Feb. 1997.

[46] E. I. Tsai, B. A. Coan, M. Kerner and M. P. Vecchi. A Comparison of Strategies for Survivable Network Design: Reconfigurable and Conventional Approaches, in *Proc. IEEE GLOBECOM*, pp. 49-55, 1990.

[47] D. B. West. *Introduction to Graph Theory*, Prentice Hall, 1996.